PHP
للمبتدئين

إعداد

محيي الباز

أرحب بالاستفسارات و الاقتراحات على

Email : mohey.elbaz@gmail.com

MSN : mohey.elbaz@hotmail.com

مقدمة

بسم الله الرحمن الرحيم و الصلاة و السلام على سيدنا محمد صلى الله عليه و سلم تسلما كبيراً ، أما بعد ، فالعالم في تطور مستمر و التكنولوجيا في عطاء بناء لكل من يسايرها و كل من يتعلم أو على الأقل يكون لديه من الفضول ما يكفي ليجعله يفكر مجرد التفكير في هذه الأمور ، لذا فمن الواجب على كل من لديه العلم أن يساعد غيره ممن يريدون التعلم على ذلك ، و ها أنا أضع بين يديك أولى الخطوات نحو PHP أسهل و أفضل إن شاء الله .

و ما أرجو منك إلا الدعاء لي و لوالدي

لمزيد من المعلومات و الاستفسارات

Email : mohey.elbaz@gmail.com

MSN : mohey.elbaz@hotmail.com

استكشاف بيئة عمل PHP

صفحات الانترنت هي في الأصل عبارة عن مجموعة من السطور البرمجية يتم تحليلها لتظهر مره أخرى للمستخدم عن عرضها على أي برنامج من برامج استعراض الويب و من أهمهم Internet Explorer ، و لغة PHP تمكنك من التحكم في صفحات الويب الخاصة بك بشكل غير ثابت بعكس صفحات Html العادية و التي تكون فيها الصفحات صفحات ثابتة أو static document .

و في إطار تعاملنا مع PHP يجب أن نهيئ بيئة العمل للتعامل مع تلك اللغة ، من برنامج محرر إلى المحلل الخاص باللغة و هذا ما يجب عليك أن تقوم بتثبيته أولا قبل البدء :

- Apache Server
- PHP
- إعدادات PHP
- PHP Editor

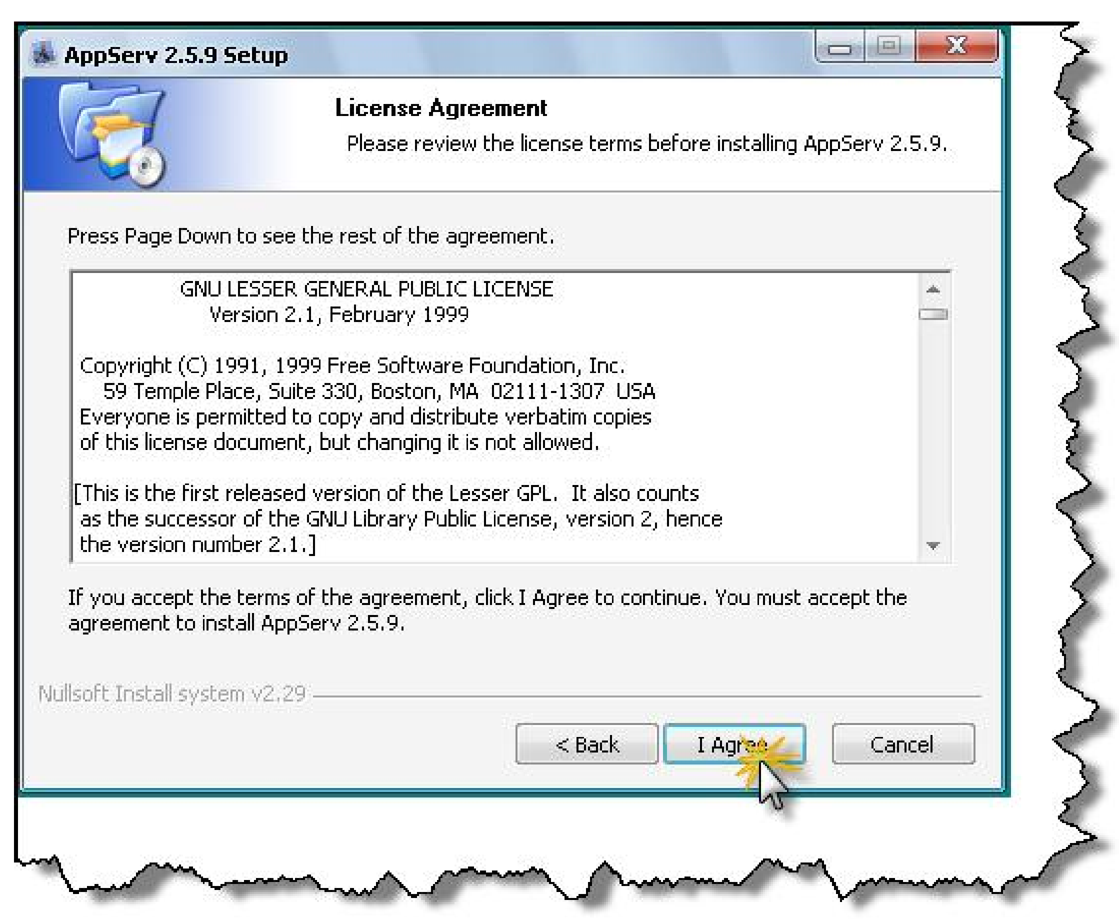
إعداد الخادم الشخصي للمرة الأولى

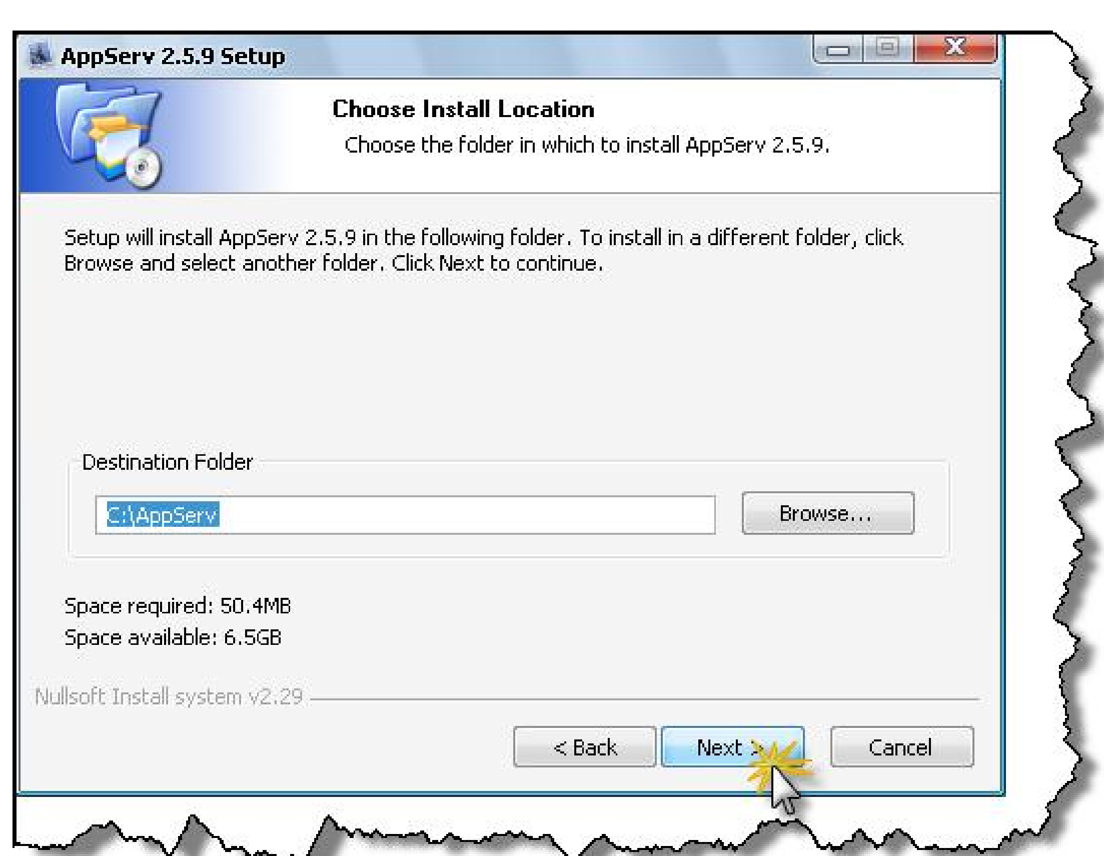
قم بالدخول على موقع www.appservnetwork.com و تحميل آخر إصدارة من برنامج AppServ ، البرنامج مزود بكل البرامج و التطبيقات و المحللات التي تحتاج أليها لبيئة العمل الخاص بك و هي :

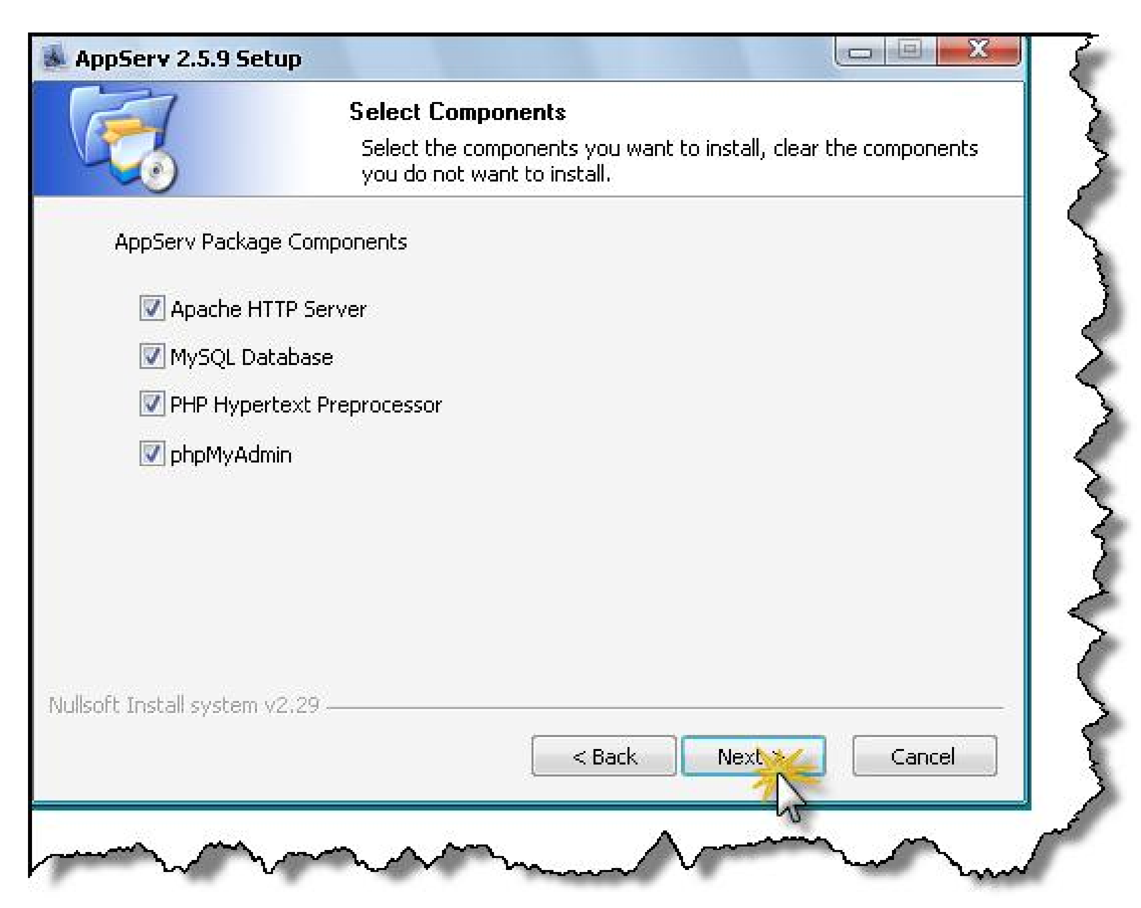
- Apache
- PHP
- MySQL
- phpMyAdmin

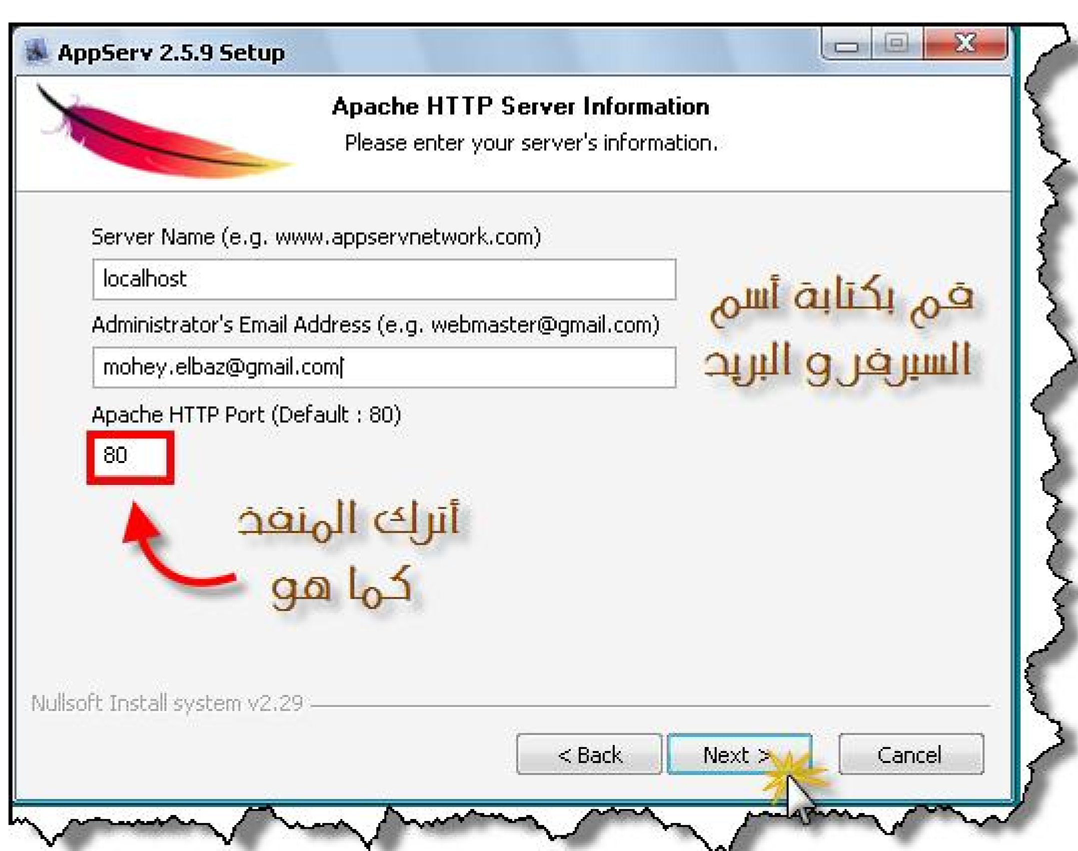
و هذه المكونات لازمه بشكل ضروري للقدرة على تكوين بيئة عمل مناسبة للتعامل مع PHP في نظام Microsoft Windows ، و الآن لنتابع عملية تركيب AppServ :

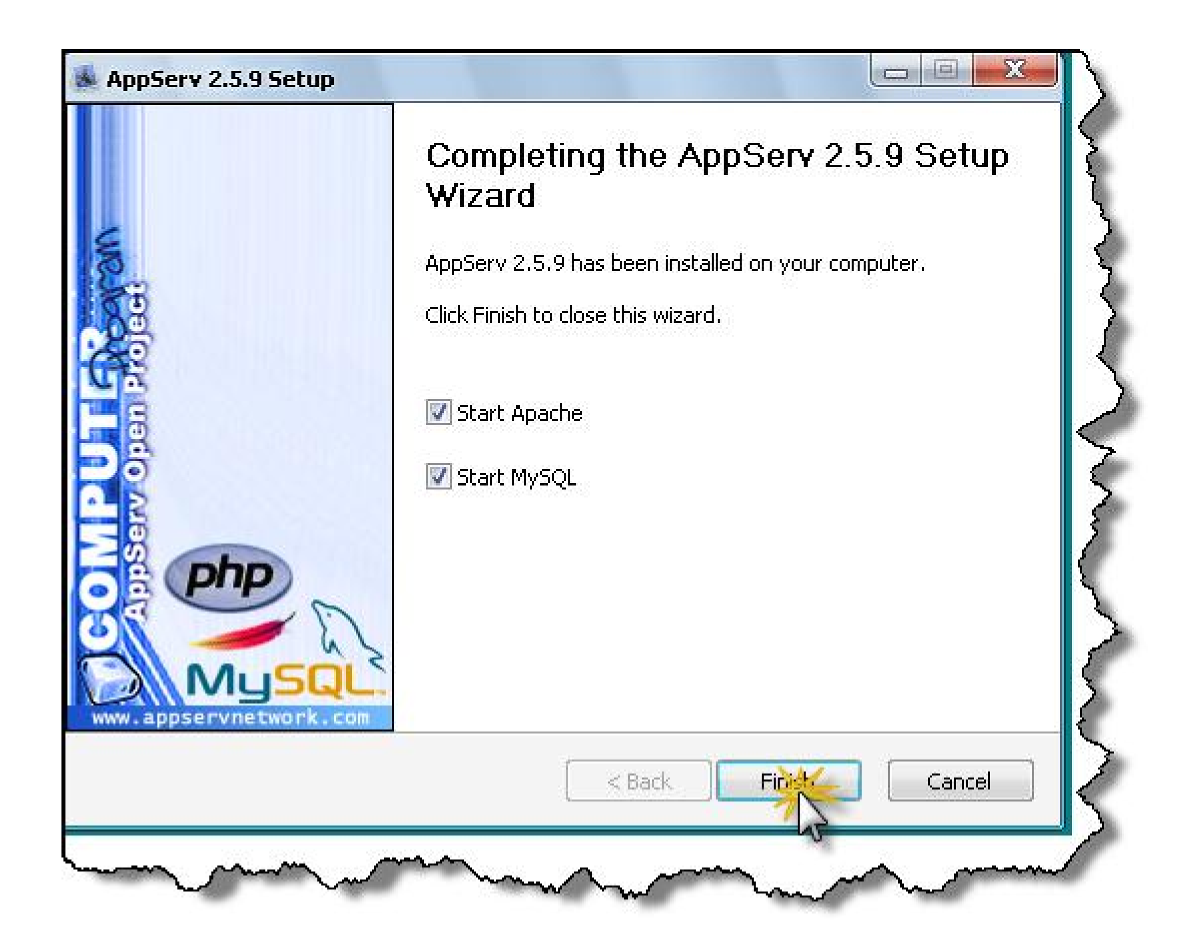



AppServ 2.5.9 Setup
License Agreement
Please review the license terms before installing AppServ 2.5.9.
Press Page Down to see the rest of the agreement.
GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999
Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install AppServ 2.5.9.
Nullsoft Install system v2.29
< Back
I Agree
Cancel


AppServ 2.5.9 Setup
Choose Install Location
Choose the folder in which to install AppServ 2.5.9.
Setup will install AppServ 2.5.9 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\AppServ
Browse...
Space required: 50.4MB
Space available: 6.5GB
Nullsoft Install system v2.29
< Back
Next >
Cancel


AppServ 2.5.9 Setup
Select Components
Select the components you want to install, clear the components you do not want to install.
AppServ Package Components
Apache HTTP Server
MySQL Database
PHP Hypertext Preprocessor
phpMyAdmin
Nullsoft Install system v2.29
< Back
Next >
Cancel


AppServ 2.5.9 Setup
Apache HTTP Server Information
Please enter your server's information.
Server Name (e.g. www.appservnetwork.com)
localhost
Administrator's Email Address (e.g. webmaster@gmail.com)
mohey.elbaz@gmail.com
Apache HTTP Port (Default : 80)
80
قم بكتابة أسم
السيرفر و البريد
أترك المنفذ
كما هو
Nullsoft Install system v2.29
< Back
Next >
Cancel

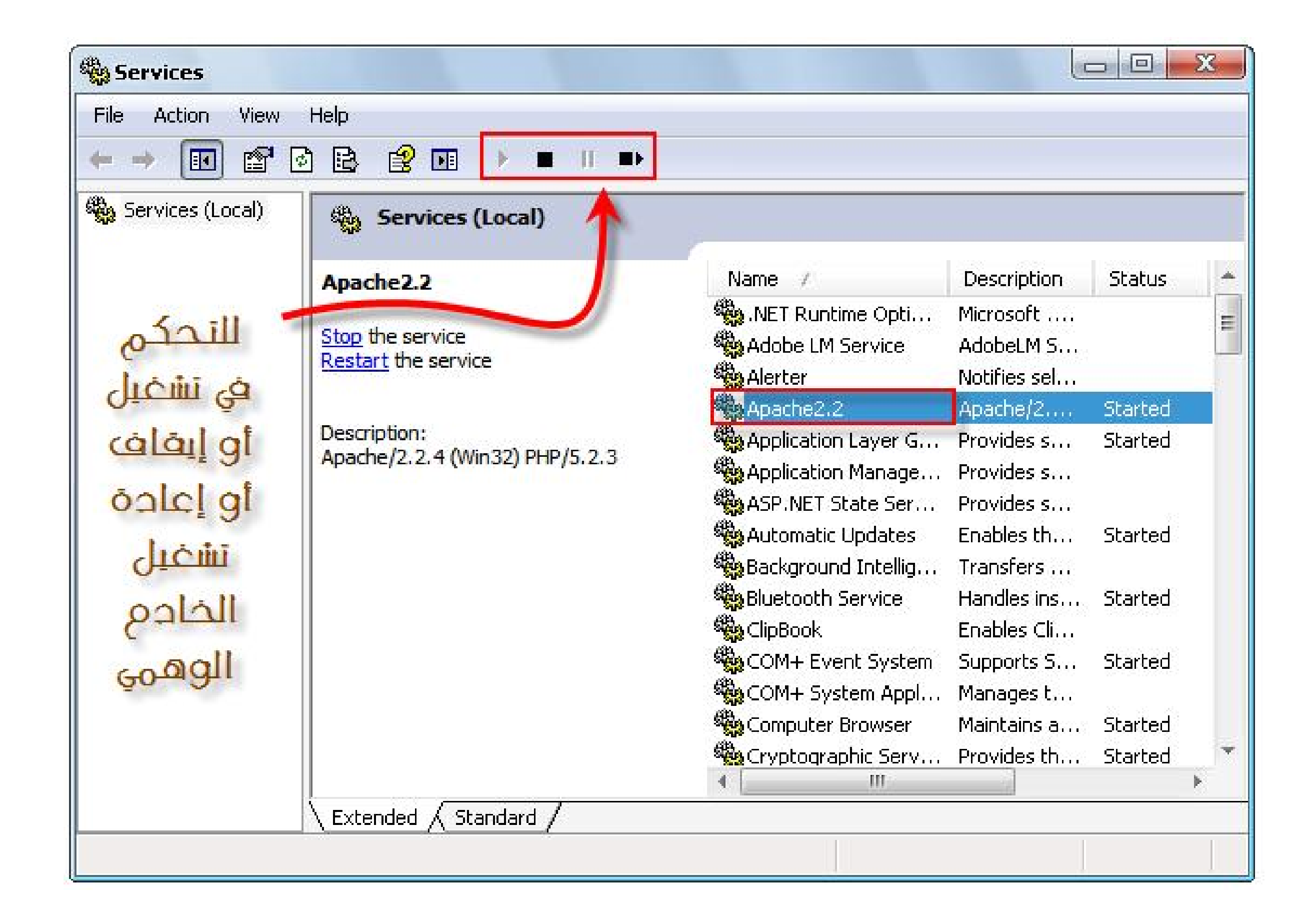


- **التحكم في الخادم الوهمي :**

الآن و قد تم تثبيت الخادم الوهمي يجب أن تقوم بالتأكد من نجاح التثبيت و عمل الخادم الوهمي، و هذا يتم عن طريق مدير الخدمات في النظام Windows XP ، و منها أيضاً يمكنك إيقاف أو تشغيل أو تعطيل أو إعادة بدأ خدمة الخادم الوهمي في النظام ، كما أنه يوجد طريقة أخرى لذلك و لك الطريقتين :

الطريقة الأولى من Windows Service :

١- من لوحة التحكم Control Panel
٢- ثم Administrative Tools
٣- ثم Services
٤- منها تجد النافذة الموجودة بالصورة
٥- قم بتحديد Apatche2.2 من القائمة مع العلم أن الإصدار قد يختلف أي أنه من المهم فقط وجود Apache
٦- عند التحديد سوف تجد الأدوات التالية ظاهرة بالأعلى و هي محددة لكل من تلك الأزرار وظيفتها حيث أنها مرتبه من اليمين إلى اليسار، الزر الأول و هو على شكل مربع جواره مثلث و هذا يقوم بعمل إعادة تشغيل للخدمة ، الزر الثاني على شكل خطين متوازيين و هو يقوم بإيقاف الخدمة مؤقتاً ، الزر الثالث و هو على شكل مربع و يقوم بإيقاف الخدمة عن العمل، الزر الرابع و الأخير يقوم بتشغيل الخدمة في حالة توقفها.
٧- لتتأكد من عمل الخدمة سوف تجد حالة من أثنين ، أما أن زر التشغيل غير مفعل و إيقاف التشغيل هو المفعل و بذلك فإن الخدمة تعمل في وقتها و إلا فإنها تكون الحالة الثانية و هي أن الخدمة لا تعمل و في هذه الحالة يجب أن تقوم بتشغيلها من الزر الرابع ناحية اليسار في المربع المحدد باللون الأحمر.



الطريقة الثانية Apache Monitor :

١- من قائمة البدا Start
٢- ثم All Programs
٣- ثم AppServ
٤- ثم Control Server by Service
٥- ثم التطبيق Apache Monitor
٦- ستجده كما بالصورة

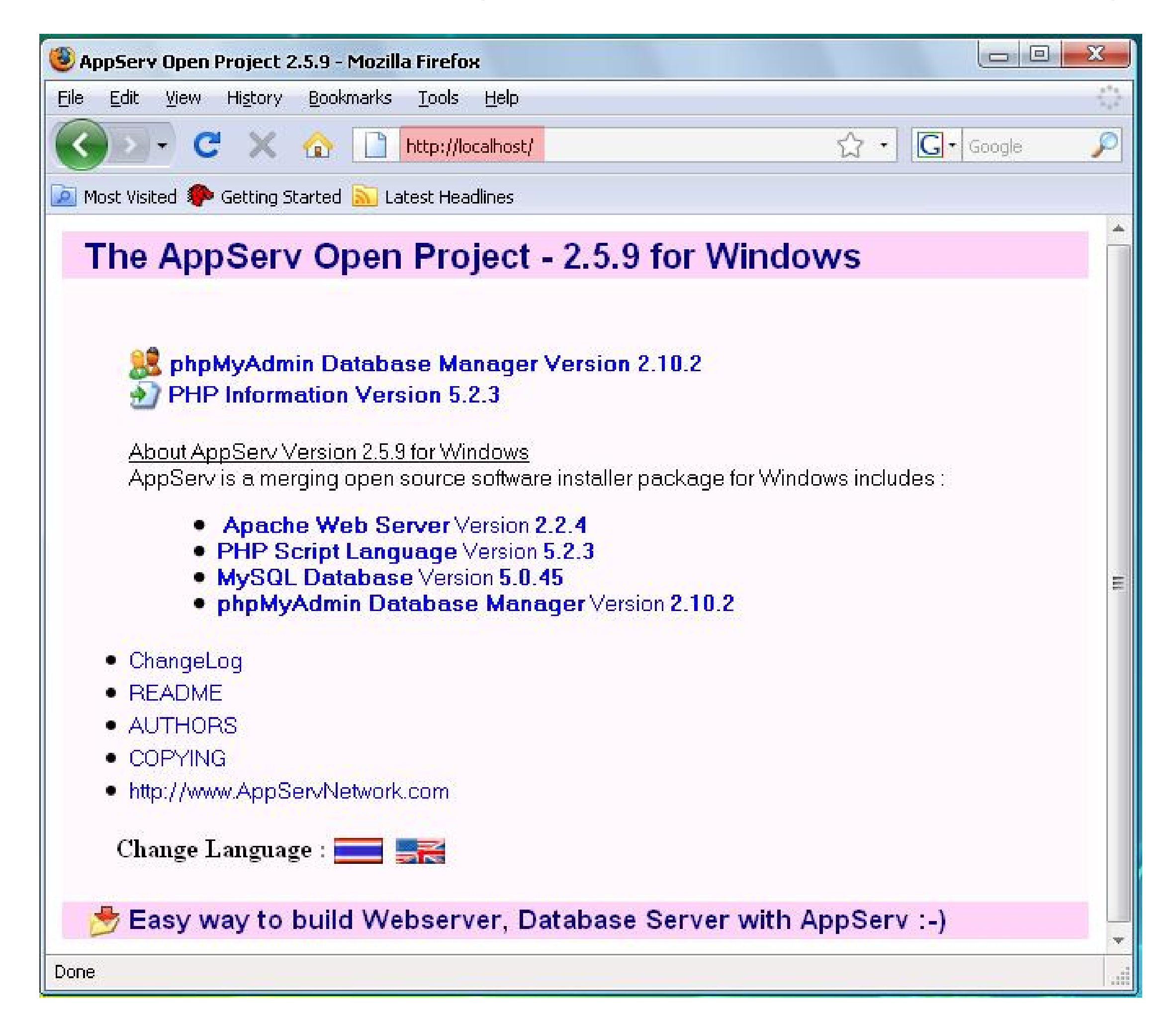


- **التأكد من عمل الخادم الوهمي :**

١- قم بفتح المتصفح الخاص بك

٢- أكتب عنوان الموقع كالتالي http://localhost

٣- إذا تم فتح الصفحة بالصورة أعلم أن خادمك المحلي يعمل بنجاح حتى الآن



بهذا تكون قد قمت بتركيب الخادم الوهمي على نظامك و كذلك التأكد من عمله بنجاح و كذلك لديك الإمكانية للتحكم في حالة عمله و في انتظار كتابه أول سطر برمجي بلغة PHP هذا و بعد أن تم التأكد من عمل المحللات اللازمة لبيئة العمل .

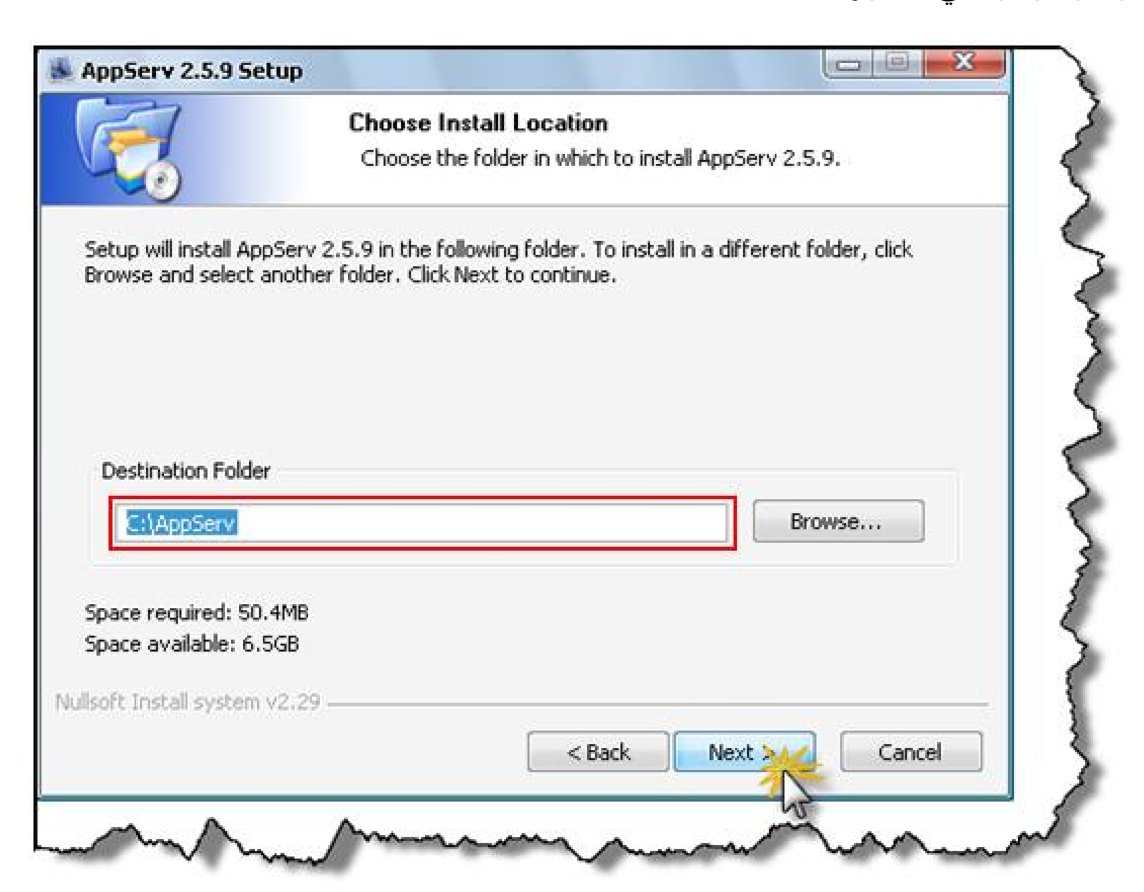
- **كيف يعمل الخادم الوهمي :**

الخادم الوهمي عبارة عن برنامج يقوم بتحويل جهازك إلى خادم محلي يخدم نفسه و الشبكة الموجود فيها . و لنتحدث داخل نطاق البرمجة ب PHP دون الدخول في تفاصيل الشبكات ، لذا فإن أي صفحة من صفحات الانترنت التي تدخل عليها تكون موجودة على خادم أو جهاز ذات مواصفات عاليه لخدمة أكبر عدد ممكن من المستخدمين ، الصفحة تكون موجودة على الخادم و عندما تقوم أنت بالنداء عليها بواسطة كتابه عنان الصفحة في المتصفح الخاص بك فإن الخادم يقوم بتحليل اللغة إلى أبسط ما يمكن و هي Html ليتم إرسال تلك اللغة إلى المتصفح يتم عرضها بالصورة المناسبة ليظهر الموقع على ما يبدو هو عليه .

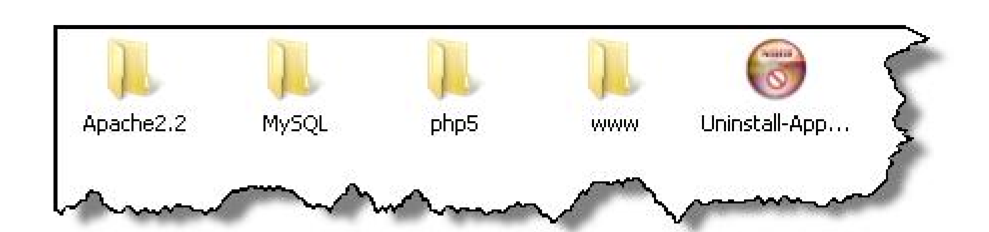
يتم الدخول على الخادم أما باسم الموقع أو أسم النطاق للموقع مثلا www.google.com أو عنوان IP الخاص بالخادم 209.85.135.103 و لكن كان من الأفضل أن يكون بالاسم حيث أن صعوبة تذكر IP كافية ليسبب وجود أسماء النطاقات أو Domains . وفي الحقيقة أسم النطاق لا يعمل إلا كوسيلة نقل ليس إلا ، أي أنه عندما تقوم بكتابه أسم النطاق مثلا www.google.com فإن هذا الاسم يتحول إلى IP 209.85.135.103 ليتم التعامل مع الخادم .

الهدف من كل هذا هو أنني أريد أن أنبهك إلى أن الجهاز الخاص بك و بعد تثبيت AppServ لديه اسم نطاق و كذلك IP و كلاهما وهمي ، و هذا واضح جداً في الصورة بالأعلى حيث أنك تجد أن عنوان الصفحة هو Localhost و لو قمت بكتابه 127.0.0.1 فسوف تجد هذه الصفحة.

و يبقى السؤال من أي يقوم الخادم الوهمي باستقبال تلك الملفات !! ، تلاحظ أنه عندما قمت بتثبيت البرنامج طلب منك عنوان التثبيت و هو موجود في الصورة بالأسفل.



إذا فنقوم بالدخول على هذا العنوان مع العلم أنني قمت بتثبته على هذا المسار [E:\AppServ] لكي لا يحدث أي لبس نتيجة اختلاف مسار التثبيت ، الآن لنقم بالدخول على المسار الذي قمت بتثبيت البرنامج فيه :



لدينا ٤ مجلدات و ملف تنفيذي واحد ، هذا الملف هو معالج إلغاء تثبيت الخادم الوهمي أي أنه إلغاء التثبيت AppServ بكل بساطه.

١. المجلد www :

و هو المجلد الذي يحتوي على الصفحات التي يتم عرضها عند الدخول على الخادم الوهمي أي أن ملف موجود داخل هذا المجلد من الممكن أن يتم عرضه من المسار http://localhost .

٢. المجلد php5 :

و هو المجلد الذي يحتوي على المحلل الخاص بلغة PHP ، أي صفحة مكتوبة بلغة PHP يتم تحليلها أولا بواسطة المحلل أو Compiler الموجود في هذا المجلد و يسمى PHP.exe، و يتم تحويل الأكواد المكتوبة من PHP إلى اللغة المبسطة و هي لغة الويب الأولية Html ليتم عرضها في المتصفح .

٣. المجلد MySQL :

و هو المجلد الذي يحتوي على المحلل الخاص بقواعد البيانات من نوع MySQL و الذي يستخدم في لغة PHP و الذي سأتناوله بالشرح فيما بعد.

٤. المجلد Apache2.2 :

و هو المجلد الذي يحتوي على البرامج الخاص بالخادم الوهمي و التي تنشئ اتصال Http عن طريق المنفذ Post ٨٠ و هو المنفذ الخاص Http.

- **إعدادات PHP :**

إن إعدادات PHP من الأمر الهام جدا و خصوصا لمبرمجي PHP حيث أن البرمجة أو المشروع الذي يعمل عليه ستكون إعداداته متعلقة بنظام إعداد PHP على النظام الذي يعمل عليه ، و هذا يؤثر بالسلب أو الإيجاب ، فقد تكون هناك بعض الإمكانيات مفتوحة أو مغلقه ، هذه الإمكانيات تتحكم بشكل كامل في الحماية الخاصة ببرمجتك ، و على سبيل المثال الخاصة Register Global و هذه الخاصية مهم جداً إغلاقها أي جعلها تساوي OFF و هذا يتم كالتالي :

١- أذهب إلى قرص النظام و لنفترض أنه (c)
٢- أذهب إلى مسار مجلد النظام (C:\Windows)
٣- قم بفتح الملف المسمى PHP.ini بواسطة برنامج Notepad
٤- قم بالضغط على (Ctrl+F) أو من القائمة Edit ثم Find
٥- قم بإدخال (register_globals) في محرك البحث دون القوسين
٦- سوف تجد السطر هكذا (register_globals = On)
٧- قم بتغيير On إلى Off
٨- قم بعمل إعادة تشغيل للخادم الوهمي بأحد الطريقتين المذكورتين مسبقاً لتطبيق الإعدادات على الخادم

جدير بالذكر أن الملف PHP.ini يوجد به التحكمات الكاملة في المحلل الرئيسي PHP.exe أي أن PHP.exe يقوم باستقبال تلك الإعدادات في كل مره يتم تشغيل الخادم فيها و لهذا تم عمل إعادة تشغيل للخادم ليحصل PHP.exe على البيانات الجديدة من PHP.ini و يتم تطبيقها على الصفحات التي يتم تحليلها.

جدير بالذكر أيضا أن الخاصية Register Global من أخطر الثغرات التي يمكن استغلالها في معظم البرمجيات القديمة و التي كانت تعتمد على الإصدارات القديمة من PHP، و يفضل في كل الأحيان لا في أغلبها الاعتماد على إغلاق تلك الخاصية.

أول برنامج لك في PHP

الآن و قد أصبح جهازك معد كليا للتعامل مع PHP، لذا فإنك الآن بصدد كتابة أول برنامج لك في PHP ، و لكن هل سألت نفسك سؤالاً !! . ما هو البرنامج المحرر الذي سوف يتم استخدامه للتغيير و لكتابه ملفات PHP !!.

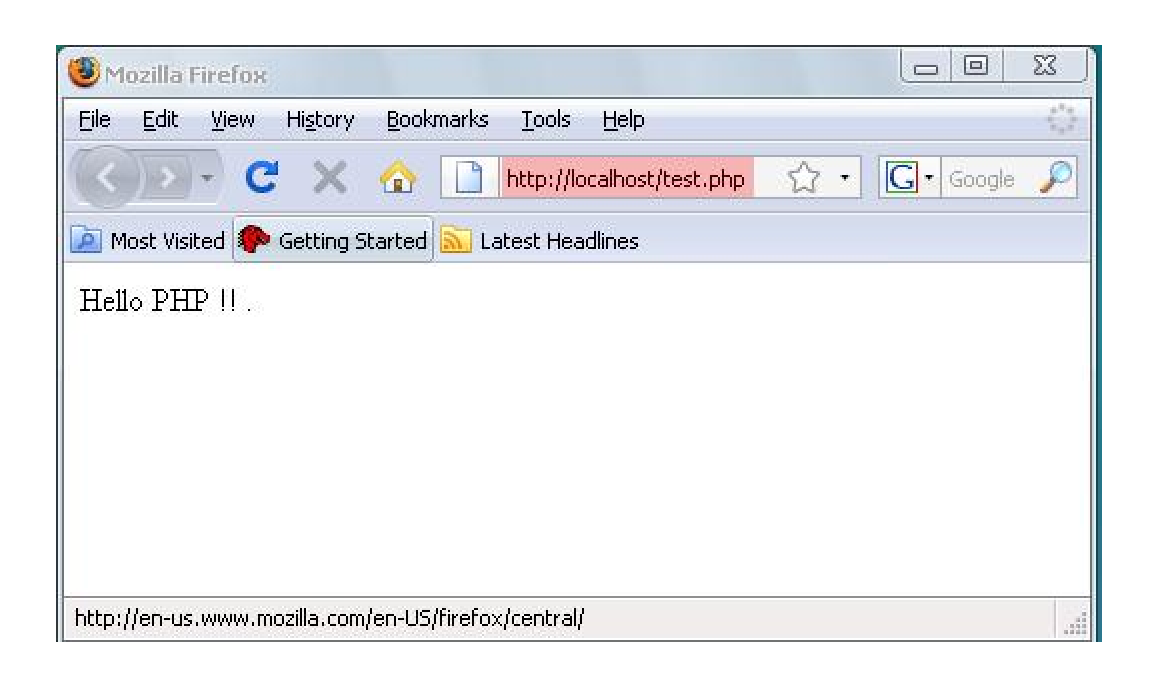
البرنامج المفضل لدى المعظم عموماً و لدي أنا خصوصا هو برنامج PHP Expert Editor برنامج مميز جداً للتعديل و الإضافة في عده من اللغات الأخرى غير PHP ، و أيضا سوف نحتاج لبرنامج آخر و هو برنامج Microsoft Front Page 2003 و هو أنسب و أفضل و أسهل برنامج للتعامل مع Html و ضبط الصفحات و تنسيقها و غير هذا.

الآن قم بتثبيت برامج التحرير الخاصة بك و إلا فإن الكتاب مرفق معك كلا البرنامجين يمكنك تثبيتهم و كذلك شرح تثبيت كل برنامج مرفق معه بالإضافة إلى شرح مبسط لكيفية التعامل مع كل برنامج.

بعد تثبيت كلا المحررين أنت مستعد تماما إن شاء الله لكتابه أول سطور برمجيه لك في لغة PHP ، الآن لتنفذ تلك الخطوات بتركيز :

١- قم بفتح برنامج PHP Expert Editor
٢- قم بإضافة صفحة PHP جديدة
٣- سيكون لديك صفحه فارغة جديدة مهيأة لكتابة أكواد PHP
٤- قم بكتابة تلك السطور
٥- بعد كتابتها قم بحفظ الملف في مجلد www باسم test
٦- قم بفتح المتصفح الخاص بك و أدخل على العنوان التالي http://localhost/test.php

```
<?php
echo "Hello PHP !! .";
?>
```



يخرج لنا المتصفح بتحليل كود الـ PHP ، لنقم بفك و تحليل الكود الموجود بالأعلى خطوه خطوة :

١- "<?php" و هذا هو وسم البداية الذي يوجد دائماً في كل صفحة برمجية مبرمجه بواسطة PHP و لا يشترط كتابة الـ " php" بعد "<?" أي انه تستطيع كتابتها بهذا الشكل "<?" فقط. و ينتهي الكود في نهاية الصفحة بـ "?>".

٢- نأتي إلى السطر الثاني ;"Hello PHP !! ." echo و هو بكل بساطه عبارة عن دالة ، و هذه الدالة تقوم بطابعة أي نص تقوم بتحديده ، نجد أنك قمت بتحديد هذا النص للعرض . !! Hello PHP و تم عرضه فعلاً ، يأتي السؤال ؟ ما معنى كلمة دالة ؟ .

الدالة هي عبارة عن سطر برمجي ، أمر يتم به إما : تنفيذ أمر ما : تنفيذ أمر ما و الحصول على المخرجات ، تنفيذ أمر ما و إضافة المدخلات ، أو تنفيذ أمر ما و الحصول على مخرجات.

و في المثال السابق قمت بإدخال نص و تنفيذ الدالة و الحصول على المخرجات في نفس الوقت أي أن الدالة بكل بساطة تعني : عبارة عن أمر برمجي يأخذ متغيرات ويعطي نواتج و لا يشترط حصوله على المتغيرات و لا حتى على إعطاء النواتج.

ملاحظات هامة قبل البدء

في أولى الخطوات لدراسة PHP يجب أن تتعرف على بعض من الأمور الهامة و التي يتم استخدامها دوماً في الأمثلة ، يجب أن تكون واعيا جداً لأي من هذه الأمور فتذكرها بالأمر الواجب.

- **التعليقات :**

```php
<?php

echo "Hello PHP 1";

//تعليق في سطر واحد

echo "Hello PHP 2";

/*
تعليق في أسطر متعددة
*/

echo "Hello PHP 3";

?>
```

التعليقات في PHP تعتبر أمر بديهي أو أمر أعتاد المبرمج أو هاوي البرمجة أو حتى المطور أو المستخدم العادي ، هذا لأنها تستخدم أحيانا لشرح شيء ما في الصفحة ، كما في الصورة تجد أن التعليق يكون لونه مختلف عن لون البرمجة حيث أنه يكون ذات لون فضي و النص مائل و هو لا يؤثر على البرمجة حيث أن ما يكب بداخل التعلق لا يتم تنفيذه برمجياً.

التعليق يكون في صورتين ، فيمكنك كتابه التعليق في سطر واحد مثلا بأن تسبق الجملة أو التعليق ب (//) ، أو أنك تقوم بكتابة (*/) قبل بداية التعليق و (/*) بعد كتابته كما ف الصورة ، لاحظ أنني قمت بتلوين الأجزاء المذكورة بالصورة.

- **نهائية السطر :**

لاحظ في الأمثلة السابقة أنه في PHP يتم إنهاء كل سطر ب (;) و هذا ليتم إعلام PHP على إنهاء السطر البرمجي و بدون هذه العلامة فإنك تمتلك خطأ في السطر البرمجي الذي ينقص هذا الرمز بنهايته، و لكن إذا كان لديك سطر برمجي واحد فلن يحدث الخطأ ، أو أنك قمت بإدخال الرمز على كل سطور الصفحة عدا السطر الأخير فلن يحدث الخطأ.

- **خطأ PHP :**

```php
1 <?php
2 echo "Hello"
3 echo "PHP"
4 ?>
```

من الأكيد أنك قد تواجه خطأ PHP ، هذا يحدث عند الخطأ في كتابة كود برمجي معين، و لنفترض كما سبق أنك لن تقوم بإدخال الرمز (;) في نهاية سطر برمجي.

```
Parse error: syntax error, unexpected T_ECHO, expecting ',' or ';' in E:\AppServ\www\man.php on line 3
```

لاحظ الخطأ التالي ، حيث قام PHP بإبلاغك بنوع الخطأ و الذي تم تحديده في بداية رسالة الخطأ و كذلك عنوان الصفحة التي بها الخطأ و المحدد باللون الأخضر و كذلك رقم السطر الموجود به الخطأ و هو السطر ٣ .

* **المسافات في PHP :**

تذكر المثال الأول ، الآن قم بكتابته مرة أخرى و لكن هذه المرة قم بعمل فواصل كبيره بين Echo و القيمة ، هل يحدث أي تغيير في تنفيذ السطور البرمجية ؟ الجواب لا بكل تأكيد ، حيث أن PHP لا يتأثر بالمسافات.

```
1  <?php
2  echo    "Hello PHP !! ."      ;
3  ?>
```

و يمكنك طباعة مسافة كبيره عن طريق الضغط على المفتاح (Tab) و المسافة العادية عن طريق الضغط على (المسطرة) كما تعلم .

* **تلوين أكواد PHP :**

تلاحظ في أكواد PHP أنها ملونه بشكل منظم جداً حيث أن الدوال لها ألونها و المتغيرات لها ألونها و النصوص لها ألونها، كل شيء ملون و مرتب بحيث يسهل عليك التعامل مع البرمجة و السطور البرمجية.

المتغيرات

المتغيرات في PHP من أولى الأمور التي يجب أن تتعرف عليها فهي ضرورية جداً في التعامل مع أي برمجة صغيرة كانت أو كبيرة ، ففي البرنامج الموجود بالأعلى كتبت (`echo "Hello PHP !! .";`) مع المتغيرات يمكن أن تجعل الكود كما يلي .

```
<?php

$value = "Hello PHP !! ."
echo $value ;

?>
```

المتغير بأبسط صورة عبارة عن أناء فارغ و لهذا سمي متغيراً ، يمكن أي تقوم بتعبئته بأي نوع من البيانات ليتم الاحتفاظ بها أثناء تنفيذ الصفحة و يتم إفراغه عند إغلاقها.

تلاحظ أن المتغيرات في PHP يتم كتابة ($) في بداية أسم المتغير و هذا لتحديده كمتغير و إلا فكيف سيتعرف المحلل علية كمتغير.

```
<?php
$أسم_المتغير = القيمة ;
?>
```

لاحظ أن : أسم المتغير حساس للحروف هذا يعني أن $Man غير المتغير $man ففي الأول حرف (M) Capital و هذا يعني أن المتغيرين مختلفين.

أسم المتغير : يمكنك كتابة المتغير بهذا الشكل ($man) أي يمكنك تسميته بأي شكل و لكن في حدود من a-z و أيضا من 0-9 حيث يمكنك أن تقوم بإدخال أرقام مع المتغيرات بهذا الشكل ($man5) و لكن و لكن لا يمكنه أن يكون رقماً فقط ($5) و يمكنك أن تدخل رمزاً واحداً فقط في أسم المتغير و هو (_) أي أنه قد يكون ($man_5).

من الممكن أن يتم تضمين المتغير كاملاً داخل علامات التنصيص كما يلي، و سوف يتم عرض الآتي (`My Value Is : Hello PHP !! .`) في المتصفح عند تشغيل الصفحة بعد تعديل الكود، أنه يتم استبدال المتغير بقيمته.

```
<?php

$value = "Hello PHP !! ."
echo "My Value Is : $value";

?>
```

المتغيرات نوعين :

- **متغير عادي :**

و هو المتغير المذكور في المثل أعلاه

- **متغير ثابت :**

```
<?php

Define ( "value", "Hello PHP !! ." );
echo value ;

?>
```

و هو عبارة عن متغير يتم تصريحه كمتغير عام على مستوى الصفحة كاملة، و الفرق بينه و بين المتغير العادي أن المتغير العادي يسبقه $ إنما المتغير الثابت يكون بدون $ فقط الاسم و ينطبق عليه ما ينطبق على المتغير العادي من حيث طبيعة الاسم حيث أنه لا يمكن أن يكون رقما فقط كما أنه لا يمكن أن يحتوي على رموز خاصة غير (_).

Function

```
define ( Name , Value );
define ( الاسم , القيمة );
```

يتم تعريف المتغير الثابت بهذا الشكل، حيث أن (`value`) هو المتغير، و يستخدم معظم الأحيان في حفظ المتغيرات الهامة ، كتحديد لنوع عملة مثلا على مستوى صفحة أو برنامج كامل، و فتح هذا المتغير مرتبط بالدالة (`Define`) أي أن هذه الدالة هي التي تقوم بفتح المتغير و تعريفه على مستوى الصفحة كاملة ، تحتاج الدالة لمدخلين و هما :

- أسم المتغير : و هو الذي يتخذ كاسم يتم النداء على المتغير و استدعائه به
- قيمة المتغير : و هي القيمة التي يتم وضعها للمتغير ليتم أستخدمها بأي شكل من الأشكال فيما بعد.

مثال : تضمين متغيرات أكثر من نوع في صفحة واحده :

```
1  <html dir="rtl">
2  <meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
3  <title>برنامج الأسعار</title>
4
5  <?php
6
7  Define ( 'mtype', "جنيه مصري" );
8
9  $product = "شاشه"   ;
10
11 echo "<h1>برنامج الأسعار</h1>";
12 echo "نوع المنتوج"." : ".$product;
13 echo "<br>";
14 echo "السعر"." : "."400"." ".mtype;
15
16 ?>
```

الصورة بالأعلى هي الكود الخاص بالمثال المطلوب و سوف أتناول الكود بالشرح سطراياً :

- السطر (١) : هو عبارة عن كود Html يقوم بتحويل الصفحة من اليمين إلى اليسار أو العكس و ذلك بواسطة تغيير قيمه Dir الموجودة تحت الوسم Html من RTL إلى LTR و العكس . حيث تجد أن RTL اختصار إلى Right To Left و كذلك LTR اختصار إلى Left To Right ، و Dir تعني اتجاه أو مسار.
- في السطر (٢) : كود Html مهم جداً و هو كود يقوم بتعريف لغة صفحة الويب بالأسفل و هذا يساعد على التحديد الأوتوماتيكي للترميز حيث أنه في أغلب الأحيان و مع متصفحات معينه يجب أن يتم تحديد هذا الـ Header، نجد أنه تحت الوسم Meta و هو وسم خاص بمعلومات صفحة الـ Html .
- السطر (٣) : و الذي يحتوي الوسم Title و هو عنوان الصفحة في شريط العنوان و كما نلاحظ أن كلمه " برنامج الأسعار" قد ظهر في المتصفح عند عرض الصفحة.
- السطر (٤) : و هو رمز بداية كود الـ PHP و الذي تم شرحه مسبقاً
- السطر (٧) : و في هذا السطر يتم تعريف متغيرنا الأول mtype على مستوى الصفحة كلها و نعطيه القيمة "جنيه مصري" أي أن mtype = "جنيه مصري" ، و هذا المتغير من نوع المتغيرات الثابتة.
- السطر (٩) : يتم فتح المتغير $product و إعطائه القيمة "شاشه".
- السطر (١١) : يتم طباعة نص و لكن النص مزود بوسم Html ، يأتي السؤال ! هل يجوز لي أن أقوم بتضمين لغة Html في المتغيرات و طباعتها في PHP ؟ الإجابة : نعم ، حيث أنه في هذا السطر قمنا بعرض النص "برنامج الأسعار" و لكن بين الوسم H1 و هذا الوسم هو وسم العناوين و H اختصارا لكلمه Heading أي رأس أو عنوان الموضوع ، و يوجد الوسم من H1 إلى H6 حيث أن H1 هو الأكبر و H6 هو الأصغر حجماً.
- السطر (١٢) : في هذا السطر نجد شيء من التعقيد نوعاً ما ، حيث أن القيمة المعطاة للدالة Echo مقسمة على مراحل ، نعم مقسمه و مضافة على بعضها فلنفترض مثلا أن لدينا الكود (Echo "He"."llo" ;) سوف تكون نتيجته مماثله للكود (Echo "Hello" ;) أي أن النقطة (.) تقوم بجمع نصين، و سيتم توضيح هذا الموضوع أكثر في موضوع آخر على وجه الخصوص ، و بهذا نجد أن المعروض ، لذا ففي المثال و في السطر الثاني من

الصفحة المعروضة في المستعرض يكون النص الموجود بين علامتي التنصيص "نوع المنتج" مضاف إليها " : " بالإضافة إلى قيمة المتغير $product و التي تساوي "شاشه" ليكون النص الظاهر في النهاية : (نوع المنتج : شاشة).

- السطر (١٣) : و في هذا السطر تم طباعة وسم Html فقط و هو
 هذا الوسم يقوم بتكون سطر جديد أسفل الكتابة مباشرة للكتابة، و يوجد طريقة أخرى و لكن هذا الوسم يقوم بأخذ سطرين و هو <p>.
- السطر (١٤) : هذا السطر البرمجي مشابه نوعاً ما للسطر رقم (١٢) و فيه يتم عرض النص و بعد ذلك يتم إضافة قيمة المتغير الثابت mtype .
- السطر (١٥) : يتم إغلاق التعامل مع كود PHP، و بهذا أكون قد انتهيت من شرح المثال، مع العلم أن التحديد باللون الأحمر الشفاف في الصورة بالأسفل هو تأثير على الصورة ليس نتيجة الكود البرمجي، فقط للتوضيح ليس إلا ، الآن .. بقى عليك أن تقوم بتنفيذ هذا المثال بطريقتك الخاصة مع إضافة أي درجات صعوبة لتقوم بتقييم نفسك.



أنواع المتغيرات

بعد أن قمت بالإطلاع على المتغير و كيفية العرض و التعامل المبسط للنوعين ، يكون الوقت كان حان لمعرفة تقسيمات أو أنواع المتغيرات العادية و هي :

- **المتغيرات الرقمية أو العددية (Integers) :**

و يتم تعريفها كمتغير رقمي بمجرد كتابتها، أي أنك لو قمت بكتابه ($value=10;) مثلا في سطر برمجي فإن Value تم تعرفيه على أنه متغير رقمي أو عددي ، و بمجرد إدخال الرقم إلى المتغير فإن لديك أمكانية كاملة للتحكم بالمتغير فيمكنك الطرح و الجميع و القسمة و الضرب إلى آخره كما في الصورة، في المثال التالي : و في السطور (3,4,5) تم فتح المتغيرات الرقمية الثلاثة ($y , $x , $val) و ابتدا من السطر ٧ حتى ١٤ تم تنفيذ العمليات الحسابية المبسطة الأربع ، ففي السطر (٧) تم جمع المتغيرين و القيمة التي تم الحصول عليها يتم تحزينها في المتغير $val بعد أن كانت قيمته تساوي صفر.

```
1  <?php
2
3  $x = 10 ;
4  $y = 2 ;
5  $val = 0 ;
6
7  $val = $x + $y ;
8  echo $val."<br>";   // 12
9  $val = $x - $y ;
10 echo $val."<br>";   // 8
11 $val = $x / $y ;
12 echo $val."<br>";   // 5
13 $val = $x * $y ;
14 echo $val."<br>";   // 20
15
16 ?>
```

أي أن المتغير $val يساوي مجموع المتغيرين $x و $y و بعدها في السطر الثامن يتم طباعة محتوى المتغير $val ، و تم تحديد قيمة كل عملية في شكل تعليقات بجوار كل عملية لتسهيل الفهم ، و هكذا بالنسبة لكل عملية من العمليات الحسابية المبسطة و المبينة في الجدول التالي :

العملية الحسابية	الرمز	مثال
جمع	+	Echo 2+2;
طرح	-	Echo 4-2;
قسمه	/	Echo 4/2;
ضرب	*	Echo 2*2;
إضافة رقم ١ على المتغير	++	$val++;
طرح رقم ١من المتغير	--	$val--;

- المتغيرات النصية (String) :

و هذا النوع من المتغيرات يقبل بتخزين الأرقام و النصوص ، أي أن العمليات الحسابية لا تجوز على هذا النوع من المتغيرات ن فلا يجوز جمع "أحمد" و "محمد" مثلا ، و لكن يمكن إضافة النصوص على بعضها البعض و الإضافة هنا لا تعني الجمع أبدا ، و للتوضيح ، نفترض أنه لديك متغير $first و متغير آخر $second و متغير أخير $var، المتغير الأول لديه القيمة "أحمد" و الثاني لديه القيمة "محمد" . إذا أردنا أن نقوم بإضافة المتغير الأول على الثاني فاحزر كيف سيكون الكود . أعتقد أني قمت بشرح ذلك مسبقا.

```php
<?php
$first = "أحمد";
$second = "محمد";
echo $first.$second;
?>
```

إذاً . إن كان الكود الذي قمت بالتفكير به كما يلي فأنك تتابع بشكل جيد و إلا فينبغي عليك التركيز أكثر و التطبيق على كل ما تعرفه من جديد عن هذه اللغة فالتطبيق يساعد على الفهم بنسبه كبيره جداً.

لذا فإنه تم استخدام الرمز " . " لإضافة متغير نصي على آخر، و كذلك يمكن استخدامه للمتغيرات الرقمية و لكن لن يكون الناتج هو مجموع المتغيرين بل المتغير الأول بجوار الثاني ، و يمكن كذلك إضافة متغير نصي على رقمي و رقمي على نصي كذلك و هذا ما أرجو أن تقوم بمحاولة تنفذه بنفسك.

- المتغير المنطقي (Boolean) :

و هذا المتغير تكون له قيمان فقط أما True أو False و يعني هذا صواب أو خطأ عامياً ، يأتي السؤال ! كيف أقوم بتصريح المتغير على أنه منطقي أو Boolean ؟.

```php
<?php
$blnt = true;
$blnf = false;
?>
```

يتم هذا بمجرد إعطائه القيمة و لكن ليس بين علامتي تنصيص كما في المتغيرات النصية أو رقم كما في المتغيرات العددية و لكن فقط إما True أو False ، أي أنك لو أردت فتح المتغير $bln يجب أعطائه قيمه من الاثنين كما يلي.

دوال المتغيرات

- **الدالة (Intval) :**

الدالة تقوم بتحويل المتغير من جميع أحواله إلى متغير صحيح، فإذا كان نصيا و يوجد به رقم يتم استخراج الرقم عن طريق الدالة فلو لديك مثلا "504Man"، عند إدخال هذا النص على الدالة المذكورة فإن الناتج يكون 504 رقماً. و إذا كان رقميا عشريا يتم تحوله إلى رقمي صحيح.

```
1  <?php
2
3  echo intval(504);       //504
4  echo "<br>";
5  echo intval(50.4);      //50
6  echo "<br>";
7  echo intval('504');     //504
8  echo "<br>";
9  echo intval('+504');    //504
10 echo "<br>";
11 echo intval('-504');    //-504
12
13 ?>
```

مثال :

في هذا المثال و بدون الحاجة إلى تحليل البرمجة بشكل سطري، فإنك تجد أنه في كل من السطور (3,5,7,9,11) قد تم تنفيذ الدالة على مجموعة من الأرقام موزعة في كل سطر، و في السطور (4,6,8,10) تم طباعة أو إضافة سطر جديد ليفصل بين كل سطر و الآخر و هو الوسم
 كما ذكرت مسبقاً ، حاول أن تقوم بتنفيذ هذا التطبيق، و ستجد أن النتائج مساوية للنتائج التي كتبت بجوار كل نداء للدالة Intval.

- **الدالة (Floatval) :**

تقوم هذه الدالة بالحصول على أو استخراج الأرقام العشرية من المتغيرات.

```
<?php

echo floatval("500.4Man");   //500.4

?>
```

مثال :

لو لديك متغيراً نصياً مثلا قيمته تساوي "4.5Mohamed" عند إدخال هذه القيمة على الدالة المذكورة فأنك تحصل على الرقم 4.5 ، في هذا المثال : عند تنفيذ الدالة Floatval يتم مسح كافة النصوص و استخراج الأرقام العشرية من النصوص، حيث أنه عند تنفيذ الكود الموجود بالصورة يتم عرض الرقم 500.4 - موجود بالتعليق – نجد أنه قام بحذف Man الموجودة بعد الرقم.

- **الدالة (Gettype) :**

تقوم هذه الدالة بإخراج قيمه هي عبارة عن نوع المتغير إما نصي String أو رقمي عددي Integer أو منطقي Boolean، و ذلك بواسطة تمرير المتغير على الدالة .

```
<?php
$intvar = 10;
echo Gettype($intvar);
?>
```

مثال :

في هذا المثال بالصورة يتم الحصول على نوع المتغير $intvar و طباعته ، و يكون الناتج Integer ، أي أن الدالة أخرجت نوع المتغير و هو Integer أي رقمي.

- **الدالة (Settype) :**

هذه الدالة تقوم بتغيير نوع المتغير الممر إليها إلى أي نوع آخر ، فلنفترض أن لديك متغير منطقي $Bol تريد أن تقوم بتحويله من نوعه الحالي إلى متغير رقمي ، كيف ذلك ! .

```
1 <?php
2 $Bol = false;
3 settype($Bol,"integer");
4 echo settype($Bol,"string");
5 ?>
```

مثال :

في هذا المثال يتم تغيير نوع واحد فقط من المتغيرات وهو المتغير المنطقي من نوعه إلى متغير نصي و رقمي و عليك بتجربة تغيير المتغير الرقمي و النصي إلى النوعين الآخرين و التجاوب مع النتائج ، الآن في السطر ٢ تم تعريف المتغير $Bol على أنته متغير منطقي يساوي True ، و في السطر ٣ تم تحويله إلى متغير رقمي و في السطر ٤ تم تحويله إلى متغير نصي ، في الحالة الأولى يكون الناتج ١ و في الحالة الثانية يكون الناتج ١ أيضا و لكن نصي ، لاحظ أن قيمه $Bol لو كانت تساوي False لكان الناتج عند تحويله إلى متغير رقمي تساوي ١ أيضاً ، فهو كرقم افتراضي عند تحويل أي متغير من عادي إلى رقمي و كذلك بالنسبة للمتغير النصي.

- **الدالة (Unset) :**

هذه الدالة تقوم بإزالة وجود المتغير من الصفحة ، تستخدم هذه الدالة في الحماية أحيانا و أحيان أخرى في تسريع عمل الصفحات حيث أن تراكم المتغيرات في الصفحة يبطئ من عملها و لكن بقدر غير ملحوظ.

```
<?php
$var = "Value";
unset($var);
echo $var;
?>
```

مثال :

في هذا المثال يتم فتح المتغير $var و إعطائه القيمة Value ، يتم تمرير المتغير على الدالة Unset ، بعد ذلك يتم طباعه المتغير ، تلاحظ ! لا شيء يتم عرضه ، هذا لأن المتغير تم إلغائه.

يوجد بعض الدوال الأخرى و لكن سوف يتم عرضها فيما بعد إن شاء الله لأنها متعلقة ببعض المواضيع الأخرى في PHP و كذلك دوال أخرى لا تهم في هذه المرحلة حيث أن الدوال التي تم شرحها.

المصفوفات

مقدمة عن المصفوفات :

إذا كنت تريد عمل كود يقوم بعرض الشهر الميلادي بعد إدخال رقمه، يعني مثلا تحويل ١٠ إلى أكتوبر ، لذا فينبغي أن يكون لديك جدول بمعنى جودل وهمي ليس جدولاً مرسوما ، هذا الجدول يحتوي على عمودين العمد الأول الشهر بالرقم و العمود الثاني الشهر بالاسم .

على كلٍ ، يجب أن يكون لديك أسماء الشهور على الأقل و بشكل مرتب و هذا كاف، و عند إذ فإنك بأمس الحاجة إلى المصفوفات ، فمن مجرد قراءة الاسم يتضح أن المصفوفة عبارة عن مصفوفة من المتغيرات التي يتم الإضافة و التعديد و التعامل معها بشكل سهل جداً.

المغيرات بشكل أسهل :

طبعاً ما ذلت تتذكر المتغيرات إن شاء الله، جيد ، الآن كيف يكون شكل المصفوفة ! ، المصفوفة عبارة عن متغير عادي أي أنها تبدأ بـ $ مثل المتغير العادي و أيضاً يتم اختيار أسم للمصفوفة كالمتغير العادي ، و لكن عند التعامل سواء في فتح المتغير أو العرض فإن الطريقة تكون مختلفة.

فعند فتح المتغير للمرة الأولى يتم كتابة المتغير بالشكل العادي مثلا $myarray و = و لكن لا يتم إعطاء المتغير القيمة العادية سواء رقم أو نص أو True أو False كما في المتغيرات، و لكن يتم مساواته بناتج الدالة Array و التي تقوم بفتح المتغير و إعطاء قيمه المصفوفة للمتغير العادي و بذلك يتم تحويله إلى مصفوفة، أعتقد أن الأمر صعب نسبياً مع الشرح الكتابي الآن تابع المثال.

المثال :

في هذا المثال تم فتح المصفوفة $myarray في السطر ٢ و تم إعطائه القيم "محمد" و "عمرو" و "مصطفى" أي أن المصفوفة $myarray تحتوي على هذه الأسماء الثلاثة، و يكون شكل المصفوفة الوهمي كما بالجدول :

```
<?php
$myarray = array("محمد","عمرو","مصطفى");
?>
```

Value	Index
محمد	0
عمرو	1
مصطفى	2

أعلم أنك تتعجب الجدول و لكن ، هذا شكل المصفوفة الوهمي حيث أنه في المثال السابق تم إنشاء ٣ متغيرات في المصفوفة كل منها يحمل القيمة المعينة و المبينة في العمود Value ، و لكن ما هو العمود index ! ، هذا العمود هو ترقيم تلقائي للمتغيرات أو القيم المنشأة ، حيث أنها تبدأ من الصفر إلى ما لا نهاية فبكل إضافة يتم زيادة الرقم ١ .

يأتي السؤال الثاني ! هل أستطيع أن أضع بنفسي قيم Index أو الـ Key ، نعم تستطيع و هذا عن طريق المثال التالي.

```php
<?php

$myarray = array();

$myarray[2]= "محمد";
$myarray[4]= "عمرو";
$myarray[6]= "مصطفى";

?>
```

لاحظ أنني قمت بعمل الآتي ، أولا قمت بتعريف المتغير $myarray على أنه مصفوفة ، و هذه المصفوفة فارغة ، و بعد ذلك قمت بفتح أو إيداع الصفوف الثلاثة ذات المفاتيح Key (٢) و (٤) و (٦) ، هذه الأرقام أدخلتها كمثال ، حيث يمكنك إدخال الرقم الذي تريد و لكن رقم صحيح ، و طريقة إيداع الصفوف سهله على ما أعتقد ، فقط قم بكتابة المتغير و بعدها [Key] مع استبدال الكلمة Key بالمفتاح الذي تريد ، و كأي متغير عادي يتم إعطائه القيمة، و هذا ما حدث في الصف (٢) و(٤) و(٦) .

الآن لحظت أنك تتعامل بسهوله مع كل صف في المصفوفة ، حيث أن التعامل معه كالتعامل تماما مع المتغيرات في ناحية إدخال و عرض البيانات، لذا يأتي سؤال جديد ! هل أستطيع أن أستخرج منه القيم ؟ نعم طبعاً و كأي متغير عادي ، في المثال الآتي سوف نستخدم الدالة Echo في طباعه قيمه كل متغير في صف منفرد.

```php
1   <?php
2   
3   $myarray = array();
4   
5   $myarray[2]= "محمد";
6   $myarray[4]= "عمرو";
7   $myarray[6]= "مصطفى";
8   
9   echo "$myarray[2]<br>$myarray[4]<br>$myarray[4]";
10  
11  ?>
```

مثال :

في هذا المثال تم فتح المصفوفة و إدخال البيانات كما أشرت من قبل و لكن الجديد أنه في الصف (٩) تم استخدام الدالة Echo لعرض محتوى صفوف معينه من المصفوفة $myarray لاحظ تم كتابه علامتي التنصيص (") و بداخلهما تم كتابه المتغيرات و بين كل متغير و متغير الوسم
 و أعتقد أنك تتذكر استخدامه من قبل في أمثلة سابقة ، تحليل السطر البرمجي أن الدالة Echo تقم بعرض المتغير الأول و قيمته "محمد" و يتم الانتقال لسطر جديد و يتم كتابة "عمرو" و بعد يتم الانتقال لسطر جديد و أخيراً كتابه "مصطفى لتظهر هكذا.



ملحوظة هامة : يمكنك أن تقوم بجعل الـ Key نص و لكن شرط أن تضعه بين علامتي التنصيص ، كهذا ($a["man"]).

دوال المصفوفات

بعد دراسة المصفوفات أرجو أن تكون ما ذلت متذكراً المغيرات و دوال المتغيرات فكلاهما مهم ، و الآن لننتقل إلى الدوال العامة المتعلقة بالمصفوفات :

- **الدالة (Print_r) :**

أسم هذه الدالة تحوير للنص Print Array أي أطبع المصفوفة ، أظن أن الأمر يكون سهل مع شخص قد يمتلك القليل من اللغة الانجليزية ، بهذه الدالة يتم طباعة المصفوفة على شكل متشعب أو هرمي.

```
1  <?php
2
3  $myarray = array();
4
5  $myarray[2]= "محمد";
6  $myarray[4]= "عمرو";
7  $myarray[6]= "مصطفى";
8
9  print("<PRE>");
10 print_r($myarray);
11 print("</PRE>");
12
13 ?>
```

مثال :

في هذا المثال يتم إضافة القيم إلى المصفوفة بعد فتحها و من ثم يتم استخدام الدالة Print_r لعرض المصفوفة كلها في شكل متشعب، لاحظ من السطر (٢) حتى السطر (٧) أعتقد أنك تفهمه جيداً و لكن من (٩) حتى (١١) فهي ما يتعلق بعرض المصفوفة على شكل هرمي ، السطر (١٠) فيه الدالة Print_r و التي تقوم بعرض المصفوفة على شكل هرمي و لكن يكون الفاصل بين كل سطر و الآخر هو \n و هذا لا يظهر في صفحات الـ Html العادية و لكن يظهر فقط في Textarea و هي الأداة الموجودة في Html لذلك فإن التاج أو الوسم (<PRE>) يقوم بتحويل الـ \n إلى
 لتبدو بهذا المنظر الموجود في الصورة ، لاحظ أن الدالة تقوم بطباعة المفتاح بجوار القيمة.

```
Array
(
    [2] => محمد
    [4] => عمرو
    [6] => مصطفى
)
```

- **الدالة (array_merge) :**

الاسم المستمد منه أسم الدالة Array Merge و هذا يعني دمج المصفوفة ، أي أن هذه الدالة تقوم بدمج مصفوفتين في واحده.

```
<?php

$ar1 = array("A","B","C");
$ar2 = array("E","F","G");

$ar=array_merge($ar1,$ar2);

print("<PRE>");
print_r($ar);
print("</PRE>");

?>
```

المثال :

لدينا المصفوفتين ($ar1) و ($ar2) و نريد دمجهما في المصفوفة ($ar) ، نقوم بمساواة المتغير ($ar) بناتج الدالة array_merge كما بالمثال لتصبح ($ar) مصفوفة تجمع صفوف المصفوفتين ($ar1) و ($ar2)، و يتم طبعتها للتأكيد باستخدام Print_r ، و تلاحظ عرض عناصر المصفوفتين معاً.

- **الدالة (array_reverse) :**

تقوم هذه الدالة بعكس ترتيب عناصر المصفوفة .

مثال :

في المثال يتم تعريف المصفوفة و إضافة العناصر بها و بعد ذلك بواسطة الدالة array_reverse يتم عكس ترتيب العناصر التي تم إضافتها في المصفوفة، سوف تلاحظ ذلك بعد تجربة تنفيذ الكود كما سبق من الأمثلة.

```php
<?php

$myarray = array("A","B","C");

$newarray = array_reverse($myarray);

print("<PRE>");
print_r($newarray);
print("</PRE>\n");

?>
```

- **الدالة (array_pop) :**

تقوم هذه الدالة بحذف آخر عنصر بالمصفوفة

المثال :

في المثال يتم تعريف المصفوفة و إضافة العناصر بها و بعد ذلك بواسطة الدالة array_php يتم حذف آخر عنصر في المصفوفة، تلاحظ طباعة كل عناصر المصفوفة عدا العنصر الأخير و قيمته (C) .

```php
<?php

$myarray = array("A","B","C");

array_pop($myarray);

print("<PRE>");
print_r($myarray);
print("</PRE>\n");

?>
```

- **الدالة (array_shift) :**

تقوم الدالة بعكس الدالة السابقة ، حيث أنها تقوم بحذف أول عنصر بالمصفوفة.

المثال :

في المثال يتم تعريف المصفوفة و إضافة العناصر بها و بعد ذلك بواسطة الدالة array_shift يتم حذف أول عنصر في المصفوفة، تلاحظ طباعة كل عناصر المصفوفة عدا العنصر الأول و قيمته (A) .

```php
<?php

$myarray = array("A","B","C");

array_shift($myarray);

print("<PRE>");
print_r($myarray);
print("</PRE>\n");

?>

```

العبارات الشرطية

PHP كلغة برمجه يتوفر بها كأي برمجة أخرى أوامر الشرط و التي تساعد على التعامل مع المستخدم ، و المقصود بالمستخدم هنا هو الشخص الذي يتعامل مع صفحة الـ php ، يوجد عدد من أوامر الشطر في PHP و هي :

- **العبارة الشرطية (IF) :**

```
IF ( حالة الشرط ) {
تحقق الشرط
}
```

أكيد أنك تعلم معنى IF بالعربية و هي (إذا) ، إذا ماذا أستفيد بالترجمة ، علك تتذكر أن IF في اللغة الانجليزية تتكون من جزأين (التحقق من الشرط) و (تنفيذ فعل الشرط) ، فنقول مثلا (لو أمطرت السماء لن أذهب فلان) هكذا تكون IF في PHP .

أعتقد الآن بأن علامات التعجب قد تكاثرت فوق راسك ، و لكن لا تكترث ، الآن لندع الأمثلة تخرج لنا ما بجعبتها، إليك بالمثال التالي :

مثال :

```
<?php

$val = 10;

if ( $val == 10 ){
    echo "تساوي";
}

?>
```

في هذا المثال أستخدم الدالة IF مرتبطة بالمتغيرات ، بكل بساطة دعني أترجم لك هذا السطر البرمجي و لا تتعجل ، بدأنا بـ IF ، و بعد ذلك تم فتح القوسين ليتم تحديد حالة الشرط ، في حالة الشرط لاحظ وجود المتغير $val - التي تساوي قيمته ١٠ – في جمله برمجية معناها ، (لو أن المتغير $val يساوي ١٠) ، و في جواب الشرط أو تنفيذ الشرط يتم طباعة كلمة يساوي باستخدام الدالة Echo.

بكل بساطه لو قدمت كانت هذه الجملة مقدمه لك بهذا الشكل : (لو أنك سعيد أحمد الله) ، فإذا كانت سعيد حقاً و هذا ما أتمناه ، فماذا ستفعل ، طبعاً سوف تقول الحمد لله ، هكذا تعمل الدالة فوق ، إذا كانت قيمه المتغير $val تساوي ١٠ أعرض كلمة "تساوي".

الأداة	الوصف
==	يساوي
<	أكبر من
>	أصغر من
<=	أكبر من أو يساوي
>=	أصغر من أو يساوي
<>	لا يساوي

لاحظ استخدام أداة المقارنة (==) للتحقق من مساواة قيمة المتغير للرقم ١٠ ، و هناك أدوات أخرى للمقارنة و هي كما بالجدول .

هذا يعني أنك تستطيع أن تتحقق من محتوى المتغير $val بشتى الطرق و هي أمامك بالجدول ، الآن أمامك الجدول و المثال بالأعلى ، ماذا تنتظر ! الآن قم بتجربة مثال لكل أداة من هذه الأدوات .

- **العبارة الشرطية (IF Else) :**

```
IF ( حالة الشرط ) {
تحقق الشرط
} Else {
عدم تحقق الشرط
}
```

هو في جزء منه كأمر الشرط IF و قد يكون هو الآخر جزء من هذا الأمر ، الآن و لنفترض أنك تشعر بالجوع ، لو وجهت لك الجملة التالية : (إذا كنت تشعر بالجوع كل و إلا أبقى جالساً) ، فماذا ستفعل ، طبعاً سوف تقوم للكل ، هذا على سبيل المثال ، إذا هذا الأمر هو متابعة للأمر IF الآن إلى المثال :

المثال :

في هذا المثال أستخدم أداة الشرط (<) ، و كالمثال السابق تماما و لكن عند ({) التي بعد تحقق الشرط لاحظ هذه الكلمة الجديد Else و التي تعني باللغة العربية و إلا ، و بعدها ({ عدم تحقق الشرط }) ، كما بالصورة، الآن قيمة $val تساوي ١٠ و في حالة الشرط أسأله إذا ما كان المتغير $val أكبر من ٢٠ ، طبعاً هو لا يساوي إذا فلن يقوم بتنفيذ فعل الشرط بل أنه سوف يتوجه مباشره لعدم تحقق الشرط.

```php
<?php

$val = 10;

if ( $val > 20 ){

        echo "أكبر";

    } Else {

        echo"أصغر";

    }

?>
```

و الترجمة المبسطة للسطور هي : إذا كانت قيمة $val أكبر من 20 إذا فقم بطباعة "أكبر" و إلا فقم بطباعة "أصغر" ، إذا كتابه السطور في المحرر و حفظ الصفحة و عرضها ، ما سوف يتم عرضه ، طبعاً كلمة "اصغر" هذا لأن ١٠ ليس اكبر من ٢٠، أي أن قيمة $val ليست أكبر من ٢٠، و ما يتم تنفيضه هو عدم تحقق الشرط و تم الإشارة إليها بالسهم الأحمر.

- **العبارة الشرطية (Switch) :**

```
Switch ( قيمة ) {
    case قيمة ;
     تنفيذ أمر ;
    break;

    case قيمة ;
     تنفيذ أمر ;
    break;
}
```

لاحظ تخطيط أو شرح العبارة الشرطية ، يتم كتابة أسم العبارة أولا ، بعد يتم طباعة القوسين كما في IF و لكن هنا يتم وضع قيمة ، أي أنك تستطيع وضع قيمة عددية نصية أو منطقية ، بعد ذلك يتم طباعة القوسين {} لإغلاق العبارة ، و بداخلهما يتم كتابة ما تراه في التخطيط ، أولا Case أي في حالة، و بعد ذلك قيمة معينه و إغلاق السطر بـ الرمز (;) ، ثم تنفيذ أمر معين ، و إيقاف Case الحالية عن العمل باستخدام Break أو الخروج منها ، و هكذا مع كل حالة .

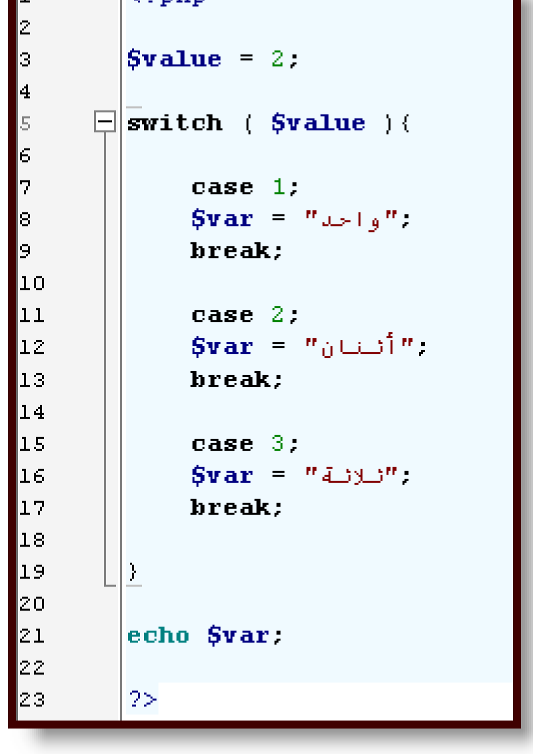
تعلمت في الموضوع السابق العبارة الشرطية IF و العبارة الشرطية IF Else و فيها كان هناك (حالة الشرط) و (حالة تحقق الشرط) و (حالة عدم تحقق الشرط)، و لكن إذا كان لديك متغير $val و تريد أن تقوم بإخراج قيمته حرفيا و ليس عدديا ، أي أن العدد (١) يسمى حرفيا واحد و العدد (٢) يسمى أثنين و هكذا ، عند استخدامك العبارة الشرطية IF لتنفيذ ذلك فسوف يكون لديك المزيد من السطور البرمجية المعقدة التي لا داعي منها مع الدالة Switch.

```php
1  <?php
2
3  $value = 2;
4
5  if ($value==1){
6      $var = "واحد";
7  }
8
9  if ($value==2){
10     $var = "أثنان";
11 }
12
13 if ($value==3){
14     $var = "ثلاثة";
15 }
16
17 echo $var;
18
19 ?>
```

مثال :

الآن فلنجرب معاً تنفيذ هذا المثال باستخدام العبارة الشرطية IF ، لاحظ في الصورة . معنا المتغير $value التي تساوي قيمته (٢) ، الآن لاحظ العبارات الشرطية الثلاث ، إذا كان المتغير $value يساوي ١ أجعل المتغير $var يساوي القيمة "واحد" ، ثم إذا كان المتغير $value يساوي ٢ أجعل المتغير $var يساوي القيمة "أثنين" ، ثم إذا كان المتغير $value يساوي ٣ أجعل المتغير $var يساوي القيمة "ثلاثة" ، و في النهاية و في السطر (١٧) قم بطباعة المتغير $var ، يتم تنفذ عبارة الشرط الموجودة بين العبارتين الأولى و الأخيرة، حيث

يتم إعطاء المتغير $var القيمة "أثنين" و بالتالي طباعته في السطر (١٧) ، حيث أن $value تساوي ٢ فعلاً، و الآن لنجرب المثال مع العبارة الشرطية Switch.



```
1  <?php
2
3  $value = 2;
4
5  switch ( $value ){
6
7      case 1;
8      $var = "واحد";
9      break;
10
11     case 2;
12     $var = "أثنان";
13     break;
14
15     case 3;
16     $var = "ثلاثة";
17     break;
18
19 }
20
21 echo $var;
22
23 ?>
```

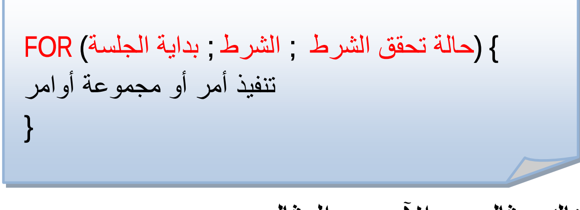
مثال :

لاحظ المتغير $value يساوي ٢ كما في المثال ت ماماً، و لكن هذه المرة نستخدم Switch لتتحقق من المتغير .

بسيطة جداً أليس كذلك ، و ترجمتها : تحقق من قيمة $value فلو أنها ١ أجعل المتغير $var يساوي ١ و أغلق الحالة ، ثم إلى الحالة التالية إذا أنه يساوي ٢ أجعل المتغير $var يساوي ٢ و أغلق الحالة و أخيراً ، إذا أنه يساوي ٣ أجعل المتغير $var يساوي ٣ و أغلق الحالة. و في النهاية طباعة محتوى $var عند تنفيذ السطور بالصورة ، ماذا سيعرض ؟ بالضبط الكلمة "اثنان".

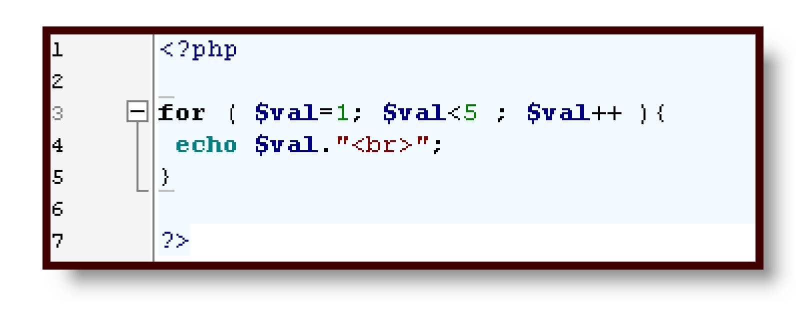
- **عبارة التكرار (For) :**

في كثير من الأحيان و في أي لغة برمجه و خصوصاً برمجة الويب تحتاج إلى عمل مجموعة من الأحداث ذات الأعداد الكبيرة ، قد تتعدى ١٠٠ و ١٠٠٠ ، طبعاً لن تقوم باستخدام العبارة الشرطية IF أو حتى Switch و لكن هنا نكون في أمس الحاجة لتدخل العبارة For ، و تخطيط تلك العبارة كما بالشكل، سيكون الأمر أسهل إذا كان هناك مثال، و الآن مع المثال .



FOR (بداية الجلسة ; الشرط ; حالة تحقق الشرط) {
تنفيذ أمر أو مجموعة أوامر
}

مثال :



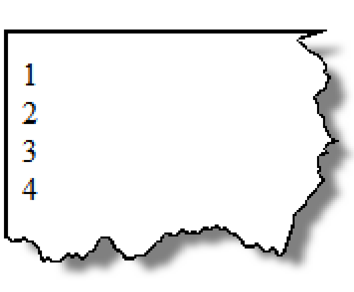
```
1  <?php
2
3  for ( $val=1; $val<5 ; $val++ ){
4   echo $val."<br>";
5  }
6
7  ?>
```

في هذا المثال ، لاحظ تم فتح المتغير و بدأ العبارة التكرارية FOR ، و كما بمخطط العبارة ، يتم الآتي :

١- فتح المتغير $val و إعطائه القيمة ١

٢- التحقق من قيمة $val لو أقل من ٥ ($val<5) نفذ حالة تحقق الشرط

٣- زيادة $val عدد ١ إضافي لتصبح قيمته من ١ إلى ٢، أي أن قيمته حاليا تساوي ٢، و هذا باستخدام (++) التي تم شرحها مسبقاً و التي في عملها تشبه : ($val = $val+1) أي إضافة واحد صحيح على المتغير .



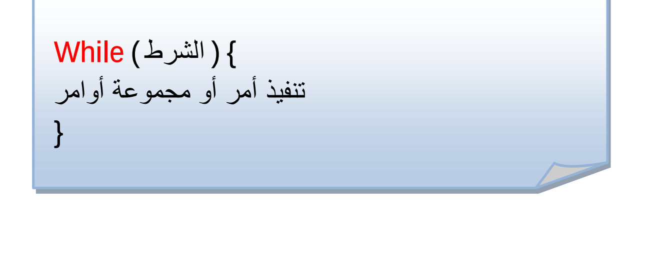
٤- يتم تنفيذ الأوامر تحت العبارة FOR و في المثال تم طباعة قيمة $val للتوضيح ، فكما تشاهد في الصورة أنه تم طباعة ١و٢و٣و٤ ، أي أنه تم تنفيذ السطر البرمجي رقم (٤) في كل مره تكون فيها قيمه $val أصغر من الخمسة .

٥- طبعاً $val=٢ و ٢ ، و الخمسة ليست أكبر من الاثنين ، لذا فسوف يتم تنفيذ الخطوات السابقة مره أخرى مع تخطي الخطوة رقم ١

٦- يتم عمل تكرار حتى تصل إلى النقطة التي تكون فيها $val تساوي ٥ ، و أكيد أن الـ ٥ ليست أصغر من ٥ ، و لكن ٥ = ٥ ، لذا فلن يتحقق الشرط و بذلك يقف عمل FOR .

أي أن عمل العبارة FOR يتوقف على تحقق الشرط و إلا فإنه يتم الخروج من العبارة FOR و الانتقال إلى السطور البرمجية الأخرى.

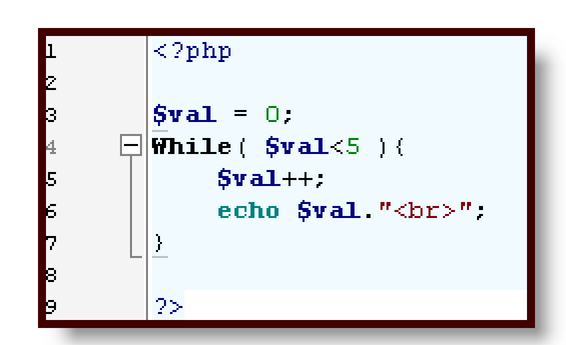
- **عبارة التكرار (While) :**



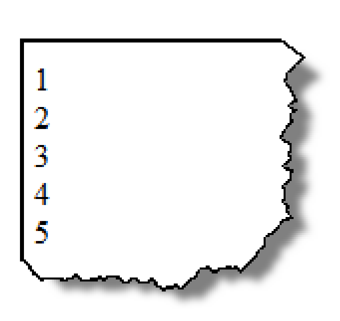
هي من أسهل عبارات و طرق التكرار في PHP حيث أنها مرتبطة بالشرط فقط ، و ترى أن كلمه While باللغة العربية تعني (بينما) ، أي بينما (يحدث كذا) {قم بتنفيذ كذا}، و هذا ما يوضحه التخطيط.

المثال :

في هذا المثال البسيط جداً و المماثل للمثال السابق في هدفه تم الآتي :



١- تم فتح المتغير $val قيمته تساوي ٠ (صفراً).
٢- طالما $val أقل من العدد ٥ نفذ الأوامر تحت While
٣- يتم تنفيذ الأوامر و من ضمن تلك الأوامر يتم زيادة المتغير $val عدد أضافي أي أن المتغير تصبح قيمته ١ ، حيث أن ٠+١=١.
٤- و بعدها طباعة قيمة $val
٥- مره أخرى يتم العودة إلى الخطوة رقم٢
٦- و يسأل ؟ هل قيمة $val أقل من ٥ ؟ الجواب : لا طبعاً فقيمته المسبقة هي ١ ، إذا يتم تنفيذ الأوامر و هكذا حتى يصل المتغير $val إلى العدد ٥ و عندها لن يتحقق الشرط و يقف عمل الدال.



و لكن تذكر أن : تنفيذ الأوامر تحت While يأتي بعد التحقق من الشرط ، و هذا ما حدث في المثال حيث أنه عندما كانت قيمه $val تساوي ٤ تم زيادة ١ في تنفيذ الأوامر و عرضه (٥) ، ليس كما هو الحال مع FOR أرجو أن تكون تلك النقطة واضحة.

كمل يجب أن تتذكر أن : يمكنك إدخال العبارات الشرطية تحت بعضها و كذلك تحت العبارات التكرارية و كذلك يمكن كتابة العبارات التكرارية تحت بعضها و كذلك تحت العبارات الشرطية، و هذا ما سوف يتم شرحه و لكن في مرحلة متقدمة.

أتوقع الآن أنك قد خطوت عدة خطوات في لغة PHP و أنته أهبة الاستعداد للدخول إلى الموضوع التالي ، و لكن أحضر و رقة و قلماً ، !! ماذا تنتظر أحضرهما فوراً ، جيد الآن قم بكتابة كل من ما يأتي سطراً تلو الآخر، و بعد كتابة كل سطر أغمض عينيك و تذكر الموضوع الذي يشير إليه السطر ، فإذا كنت تتذكره جيداً علم بالقلم عليه و إلا فعد إليه مرة أخرى لتأكيد فهمه :

١- إغلاق و تشغيل و إعادة تشغيل الخادم الوهمي.
٢- المتغيرات و أنواعها و الدوال العاملة المستخدمة مع كل نوع.
٣- المصفوفات و الدال المستخدمة في التعامل معها.
٤- عبارات الشرط.
٥- عبارات التكرار.

و أتمنى من الله أن تكون هذه الموضوعات راسخة في ذهنك و إن كان كذلك ، إذا لننتقل إلى المرحلة التالية.

الوقت و التاريخ

الوقت و التاريخ ، من الأمور الهامة جداً فقد لا تخلو برمجة من احتياجها لدوال الوقت و التاريخ ، فتسجيل عنوان موضوع بمنتدى أو انتهاء مدة إعلان معين أو عرض تاريخ اليوم بآخر الصفحة كل هذا مرتبط بدوال الوقت و التاريخ .

- **ما هو Time Stamp :**

تعلم جيداً أن جاهز الكمبيوتر الخاص بك له توقيت ، أي أن اللوحة الأم خاصتك لديها توقيت هذا التوقيت يعمل باستمرار و ساعة الحائط كذلك لها توقيتها و أيضاً خوادم الويب كذلك لديها التوقيت الخاص بها ، و هذا ما يتم الحصول عليه باستخدام دوال PHP ، و معظم هذه الدوال تعتمد على Time Stamp فما معني هذا ؟

الجواب : Time Stamp هو عبارة عن التوقيت المعتمد من قبل PHP و المستخدمة دوماً في خوادم Unix ، فلو أنك استخرجت Time Stamp الآن ، فسوف تحصل على عدد الثواني من تاريخ ١٩٧٠/١/١ الساعة ٠٠:٠٠:٠٠ حتى الآن، لذا فسوف يكون رقما كبيراً نوعاً ما .

مثال :

قم بفتح المحرر لديك و أكتب السطور الآتية الموجودة بالمثال على ما ستحصل ؟ ، الجواب رقم كبير مثل هذا (١٢٢٢٣٥٠٢٦٩) هذا كما وضحت مسبقاً Time Stamp . أي أن الدالة Time تقوم بإخراج Time Stamp الوقت الحالي.

```php
<?php

echo time();

?>
```

- **عرض التاريخ و الوقت الآن :**

الآن و بعد إطلاعك على الدالة Time و التي تقوم بإخراج وقت الخادم في صورة Time Stamp نتعرف الآن على الدالة Date و التي تقوم بإخراج التاريخ في صورته العادية الغير مشفره ، هذه الدالة في أبسط صورة تحتاج إلى قيمة ، هذه القيمة تحتوي على صياغة التاريخ أو الوقت الذي تريده ، و في الجدول في الصفحة التالية أوضح لك كل قيم صياغة الوقت و التاريخ التي تحتاج إليها .

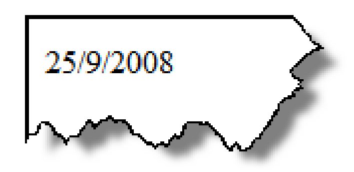
مثال :

```php
<?php

echo date("j/n/Y");

?>
```

في هذا المثال يتم استخدام الدالة Date و التي تقوم بإخراج الوقت الحالي حسب الصيغة المطلوبة و هنا الصيغة التاريخ بالأيام و بدون أصفار لتكون هكذا (1/9/1986) ، لاحظ الصيغة المدخلة للدالة هي (j/n/Y) ، و بهذا تقوم الدالة باستبدال كل رمز بقيمته من التاريخ ليكون الناتج كما بالصورة ، قم بتجربة الرموز الموجودة بالجدول .



- **التحويل من Time Stamp إلى تاريخ عادي :**

يمكن تحويل التاريخ من الصيغة Time Stamp إلى الصيغة العادية و ذلك بطريقتين :

أولا : الدالة (Date) :

الآن يمكنك عرض التاريخ أو الوقت أو أي بيانات متعلقة بالتاريخ أو الوقت ، و لكن ماذا عن Time Stamp ؟ ، لم أهملها و لكن هذا دورها ، و يأتي السؤال الثاني ! هل يمكن تحويل الـ Time Stamp غلى صيغة عادية يمكن فهمها ، نعم و باستخدام الدالة Date التي استخدمت مسبقاً في عرض التاريخ و لكن هذه المرة بطريقة مختلفة .

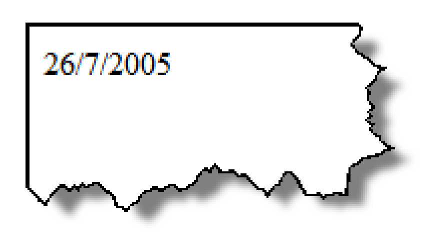
```
<?php

echo date("j/n/Y",1122350269);

?>
```

مثال :

في هذا المثل يتم استخدام الدالة Date لتقوم بتحويل التاريخ من Time Stamp إلى الصيغة العادية ، و أيضاً يتم عرض التاريخ أو الوقت حسب الصيغة الممررة إلى الدالة كما يلي و يكون الناتج كما بالصورة.



لاحظ أنها قامت بتحويل التاريخ من صورته Time Stamp إلى الصورة العادية و حسب الصيغة الممررة إلى الدالة Date.

ثانياً : الدالة (Getdate) :

باستخدام هذه الدالة يتم تحويل التاريخ و الوقت من الصيغة Time Stamp إلى الصيغة العادية ، و تكون مخرجات هذه الدالة على شكل مصفوفة تحتوي على عناصر التاريخ و الوقت أي (الثانية و الدقيقة و الساعة و اليوم و الشهر و السنة) ، فلنجرب معاً الدالة Getdate.

```
<?php

$timearray = getdate(1122350269);

echo "<PRE>";
print_r($timearray);
echo "</PRE>";

?>
```

مثال :

في هذا المثال يتم تحويل الرقم (١١٢٢٣٥٠٢٦٩) غلى الصيغة العادية من التاريخ باستخدام الدالة Getdate و يتم وضع مخرجات الدالة في المتغير $timearray ليتحول إلى مصفوفة تحتوي بيانات التاريخ و الوقت المستمد من الرقم، و في النهاية يتم استخدام الدالة Print_r لطباعة المصفوفة ، و هذا يعني أنه يمكنك استخراج اليوم أو السنة أو أي عنصر من المصفوفة كما تم شرح ذلك مسبقاً.

```
Array
(
    [seconds] => 49
    [minutes] => 57
    [hours] => 6
    [mday] => 26
    [wday] => 2
    [mon] => 7
    [year] => 2005
    [yday] => 206
    [weekday] => Tuesday
    [month] => July
    [0] => 1122350269
)
```

- **التحويل من تاريخ عادي إلى Time Stamp :**

و الآن هل يمكن التحويل من التوقيت أو التاريخ العادي إلى Time Stamp ؟ نعم يمكن ذلك عن طريق الدالة Mktime و هذه الدالة تطلب عدد من المدخلات و هي (السنة و الشهر و اليوم و الساعة و الدقيقة و الثانية) .

```
<?php
Mktime(ساعة, دقيقة, ثانية, شهر, يوم, سنة);
?>
```

الرمز	الوصف	مثال على المخرج
a	الوقت صباحا أو مساء Small	am , pm
A	الوقت صباحا أو مساء Capital	AM , PM
d	رقم اليوم في الشهر يبدأ بصفر	من ٠١ إلى ٣١
D	أسم اليوم بشكل مختصر	Sun , Mon
F	اسم الشهر بدون اختصار	August
g	رقم الساعة الآن بصيغة ١٢ ساعة لا يبدأ بصفر	1-12
G	رقم الساعة الآن بصيغة ٢٤ ساعة لا يبدأ بصفر	0-23
h	رقم الساعة الآن بصيغة ١٢ ساعة يبدأ بصفر	01-12
H	رقم الساعة الآن بصيغة ٢٤ ساعة يبدأ بصفر	00-23
i	عدد الدقائق في الساعة	00-59
I	تحديد النهار و المساء على شكل رقمين ٠أو١	١ صباح و٢ مساء
j	رقم اليوم في الشهر و لا يبدأ بصفر	1-31
l	اسم اليوم كاملاً بدون اختصار	Friday
L	حالة السنة كبيسة أو لا	١ كبيسة و ٢ غير كبيسة
m	رقم الشهر في السنة و يبدأ بصفر	01-12
M	أسم الشهر في السنة مختصر	Jan
n	الشهر في السنة على شكل رقم لا يبدأ بصفر	1-12
s	الثواني في الدقيقة على شكل رقمين	00-59
S	اسم اليوم مختصر	TH , ST , ND
t	عدد الأيام في الشهر	28-31
U	Time Stamp	1222352049
w	اليوم من الأسبوع على شكل رقم	0-7
y	السنة على شكل رقم مكون من رقمين	88,89,90
Y	السنة على شكل رقم مكون من أربع أرقام	2007,2008
z	اليوم في السنة على شكل أرقام	0-365

إرسال و استقبال البيانات

- **أنواع طرق إرسال البيانات :**

أتمنى أن يكون بدايتك لدراسة النماذج بهذه السرعة دالاً على أنك قد فهمت ما سبق جيداً ، حيث أنك الآن بصدد أهم موضوع في لغات برمجة الويب ، حيث أن هذه النقطة من أهم النقط على الإطلاق من وجهة نظري ، فلا تخلو برمجة من إرسال بيانات من صفحة إلى أخرى ، و أكبر مثال لذلك : في صفحة أتصل بنا ، تلاحظ أنك تكتب البيانات و بعد كتابتها تنقر إرسال ، فيتم تحميل صفحة أخرى ، في هذا الوقت يتم إرسال البيانات التي قمت بكتابتها من الصفحة الأولى إلى الأخرى ، فإذا كتبت مثلا (البريد و الهاتف و عنوان الرسالة و الرسالة) فبعد الضغط على إرسال سيتم نقل كل هذه المعلومات إلى الصفحة الأخرى ليتم التعامل معها ، لاحظ هذه الصورة :



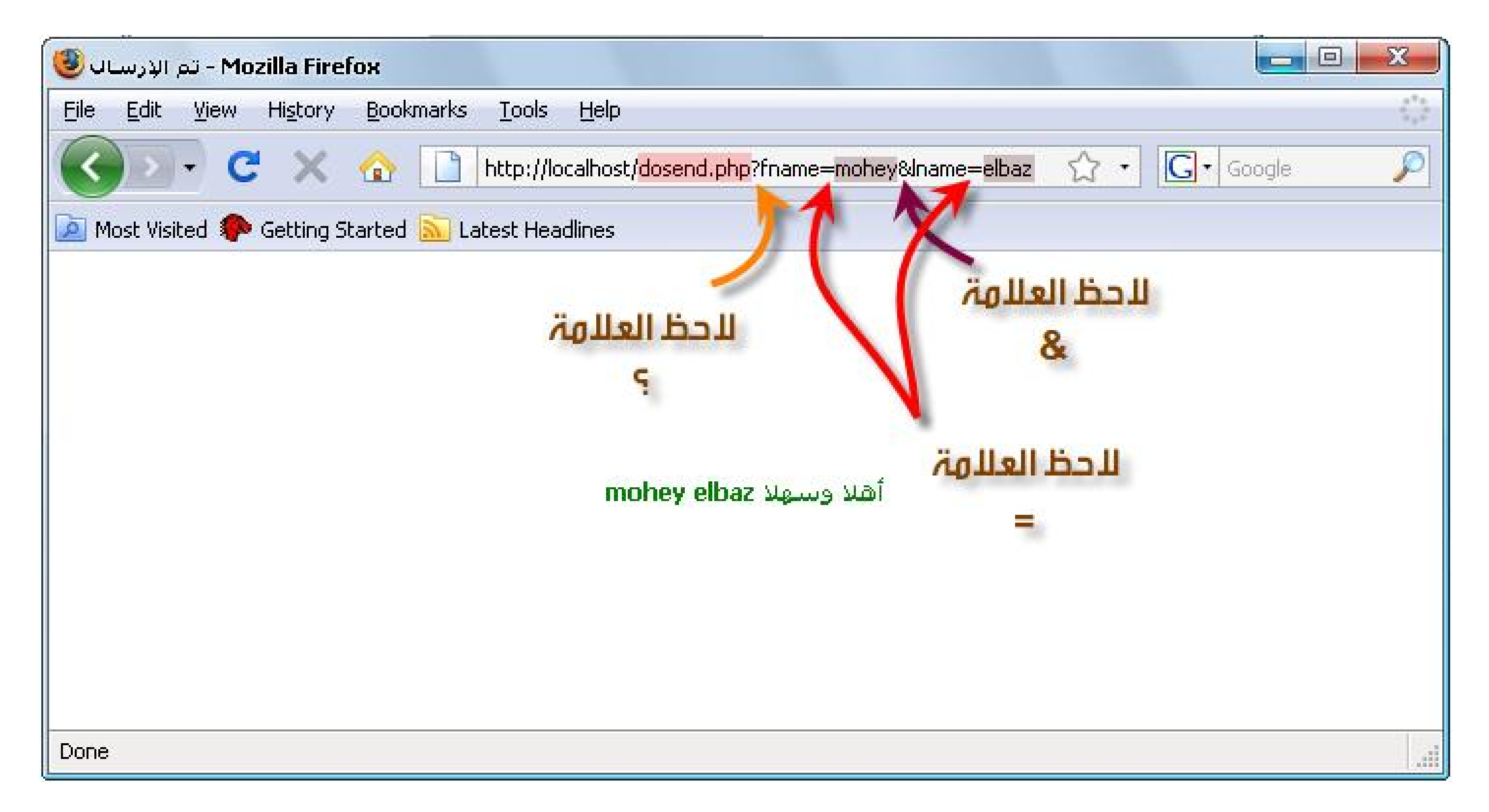
هذه الصورة كمثال على أحد نماذج الإرسال و المستخدمة في كثير من المواقع ، لاحظ عنوان الصفحة بالأعلى ، و لاحظ العنوان في الصورة الآتية :



في الصورة الأولى كان العنوان أسم الصفحة form.htm و بعد الإرسال كانت dosend.php ، و هذا ما عنيته بنقل البيانات ، حيث أن البيانات التي كتبتها تم نقلها من الصفحة form.htm غلى الصفحة dosend.php ليتم التعامل معها بأي طريقة كانت ، و نلاحظ أنك لم ترى أي من تلك البيانات ، هذا لأن هذا النوع من إرسال البيانات يقوم بإرسالها بشكي غير مرئي ، أي لا يستطيع المستخدم رؤية البيانات ، و هذا النوع من الإرسال يسمى POST ، الآن فلتلاحظ معي هذه الصورة :



هذه صورة لنموذج مبسط كمثال فقط ، يقوم بإرسال أسم المستخدم من صفحة إلى أخرى ، لاحظ صفحة استقبال البيانات و لاحظ الأسهم و النصوص التي تم تحديدها جيداً.



و في هذه الصورة تلاحظ أن عنوان الصفحة غريب نوعاً ما ، فليس أسم الصفحة فقط كما في طريقة الإرسال الأولى و لكن متضمنا بعض الأسماء و الرموز ، هذه الأسماء و الرموز عبارة عن البيانات التي تم نقلها من الصفحة form.htm إلى التالية و الموجودة بالصورة dosend.php ، و هذا النوع بعكس النوع الأول يتم نقل البيانات فيه بشكل مرئي ، أي أن المستخدم يستطيع رؤية البيانات المرسلة، هذا النوع من الإرسال يسمى GET.

الآن و بعد أن اطلعت على المثالين السابقين ، ماذا استنتجت ؟ ، نعم صحيح ، أن هناك نوعين من طرق نقل البيانات ليتم نقل البيانات من صفحة إلى أخرى :

- **الطريقة (POST) :**

و هي طريقة تستخدم لنقل البيانات من صفحة إلى أخرى بشكل مخفي عن المستخدم ، تستخدم في معظم نماذج الإرسال التي تحتوي على بيانات لا يجب على المستخدم رؤيتها ، كنماذج الشراء بالبطاقات الائتمانية و نماذج التسجيل في المواقع و نماذج إرسال البيانات و كذلك النماذج التي تحتوى على كم كبير من المعلومات.

- **الطريقة (GET) :**

و هذه الطريقة يتم بها إرسال البيانات من صفحة إلى أخرى و لكن بشكل مرئي للمستخدم و تستخدم عادة في نماذج البحث و نماذج أخرى ، و هذه الطريقة يتم إرفاق البيانات فيها برابط الصفحة المرسل إليها البيانات و هذا ما لاحظته من المثال بالأعلى ، فكان عنوان الصفحة (dosend.php?fname=mohey&lname=elbaz) ، لاحظ التلوين و تمييز العنوان جيداً ، كالمعتاد أسم الصفحة هو dosend.php ، جيد ، فلننتقل إلى الرمز التالي ، (؟) علامة الاستفهام ، هذه العلامة تهيئ الصفحة لاستقبال البيانات بالطريقة GET ، لاحظ ما بعد (؟) ، (fname=mohey&lname=elbaz) ، و هذا عبارة عن البيانات المرسلة و لكن بطريقة شبه مركبه ، بالنسبة لـ (fname=mohey) و هذا يعني أن المتغير (fname) يساوي القيمة (mohey) ، أما الرمز (&) فهو ليتم إضافة متغير آخر للصفحة، أما بالنسبة إلى (lname=elbaz) فهذا يعني أن المتغير (lname) يساوي (elbaz) .

يأتي هنا السؤال : ، ماذا لو أردت أن تقوم بإضافة متغير آخر على المتغيرين الموجودين في الرابط بالأعلى ؟ ، نعم صحيح ، جيد ، أجل سوف يتم إضافة الرمز (&) فاسم المتغير ثم الرمز (=) و قيمة المتغير بعد ذلك.

يأتي السؤال الثاني : لو أنك تريد أن ترسل متغير واحد لصفحة ما ، فهل سوف يتم استخدام الرمز (&) ؟ ، أجل ، جواب صحيح ، لن يتم استخدامها طبعاً حيث أنه لديك متغير واحد فقط فما الحاجة غليه إذا.

• ما هو النموذج :

النموذج هو عبارة عن أداة Html التي تقوم بإرسال البيانات من صفحة إلى صفحة أخرى ، تعلم أن الـ Html مكونه من أوسمه أو Tags و وسم النموذج في Html يسمى From أي نموذج باللغة العربية ، و الآن لدينا صفحة فارغة جديدة Html ، كيف أقوم بإضافة النموذج بها ؟ سؤال جيد ! ، الجواب : قم معي بالخطوات التالية :

١- أفتح البرنامج التحرير الخاص بك و يفضل أن يكون Front Page
٢- أذهب إلى صفحة كتابة الكود و أكتب ما يلي في الصورة

```
<form name="frmsend" method="POST" action="dosend.php">

</form>
```

٣- هذا هو شكل النموذج في Html و النموذج أو الـ From يحتاج إلى إمداده ببعض البيانات و هي الملونة باللون الأخضر في الصورة ، حيث أن كل منهم مهم.

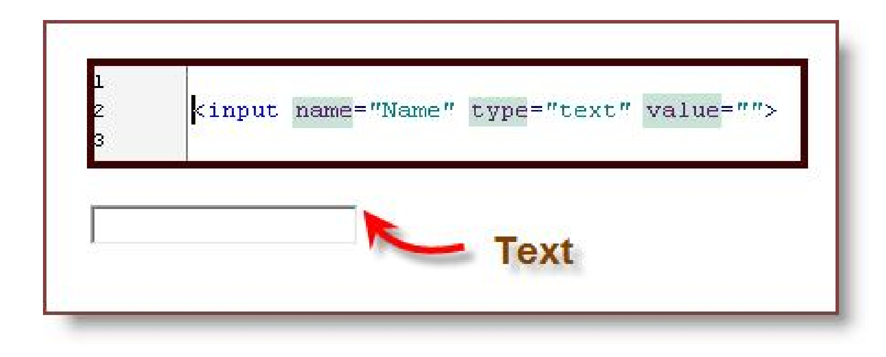
- Name : و هو أسم النموذج في الصفحة.
- Method : و هي طريقة إرسال البيانات Get أو Post.
- Action : و هي عنوان الصفحة التي يتم إرسال البيانات إليها.

٤- بهذا يكون قد تم إضافة النموذج و لكن فارغاً ، فلو حفظت الصفحة التي قمت بكتابتها و عرضها في المتصفح فلن يتم عرض شيء حيث أن النموذج فارغ ليس به أي أدوات.

• أدوات النموذج :

أي نموذج في Html لا يخلو من الأدوات و هذه الأدوات أما أن تكون ظاهره أو غير ظاهره ، و كما في الأمثلة السابقة أو حتى في بعض صفحات الإنترنت ، تجد مربع فارغ لتكتب به اسم الدخول و كلمة المرور ، تجد أزار الإرسال تجد مساحات إدخال النصوص كما في نموذج أتصل بنا أو راسلنا ، أسرد لك تلك الأدوات كما يلي :

١. مربع النص (text) :


```
<input name="Name" type="text" value="">
```


هو عبارة عن أداة إدخال النصوص و لكن في سطر واحد فقط و بشكل محدود ، كما في الصورة ، البيانات التي يحتاجها هذا الـ Tag هي :

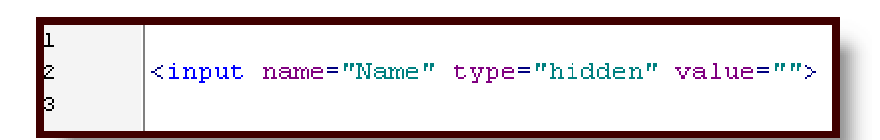
- Name : أسم المربع و هذا مهم جداً و أرجو أن تتذكره جيداً.
- Type : و هو نوع الـ Input أي

تعرفه كمربع نص

- Value : و هي قيمة المربع النصي الافتراضية ، أي أن أي قيمة تكتبها فيه سيتم عرضها في مربع النص كنص افتراضي ، لاحظ أن مربع النص في الصورة فارغ حيث أن قيمة Value = "" ، أي فارغ ، و هذا أيضاً مهما فحاول أن تتذكره بشده.

٢. مربع نص مخفي (hidden) :

هي عبارة عن أداة تساعد في إدخال النصوص و لكن بشكل مخفي ، حيث أنها لا تظهر للمستخدم و لكن يتم إرسال بياناتها كما غيرها من الأدوات.

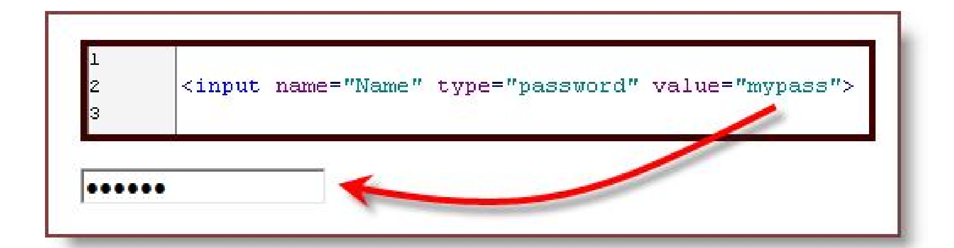


```
<input name="Name" type="hidden" value="">
```

- Name : أسم الأداة.
- Type : و هو نوع الـ Input أي تعرفه كمربع نص مخفي
- Value : و هي قيمة المربع النصي المخفي الافتراضية ، أي أن أي قيمة تكتبها فيه ستصبح كنص افتراضي .

٣. مربع كلمة المرور (password) :

هي أداة تساعد في إدخال النصوص و لكن هنا يتم تحويلها إلى الرمز (*) لكي لا تبدو للمستخدم ، و تبقى القيمة كما هي ، و هذه الأداة تستخدم معظم الأحيان في البيانات المهمة التي لا يجب أن يعلمها سوى المستخدم ، فو أنك تجلس بجوار صديقك على الجهاز الخص بك و أردت أن تدخل على بريدك الالكتروني، و لم تكن هذه الأداة متوفرة ، فسيتم عرض كلمة المرور الخاص بك بكل بساطه ، و لكن بها تكون على شكل (*) كما ذكرت.



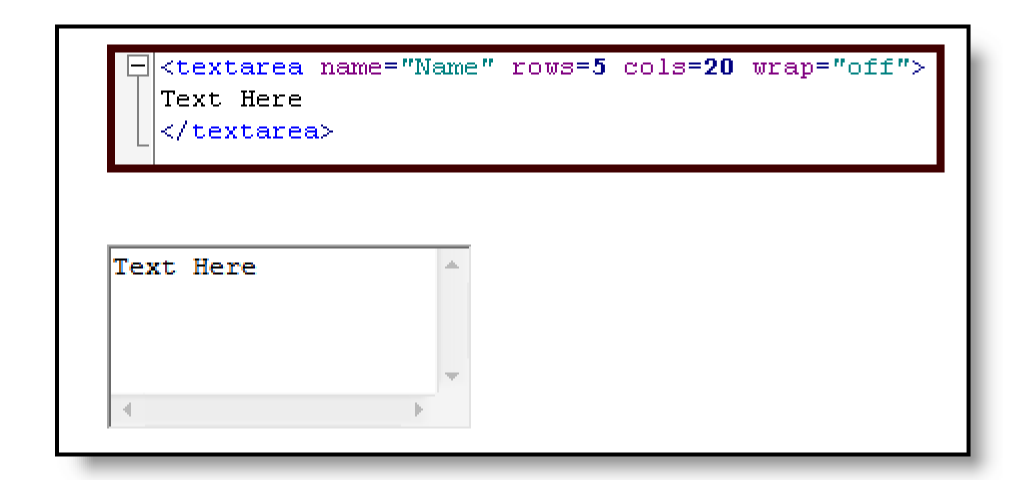
```
<input name="Name" type="password" value="mypass">
```


- Name : أسم الأداة .
- Type : و هو نوع الـ Input أي تعرفه كمربع نص لكلمة المرور.
- Value : و هي قيمة مربع كلمة المرور الافتراضية ، أي أن أي قيمة تكتبها فيه ستصبح كنص افتراضي .

٤. مساحة إدخال النصوص (textarea) :

أداة تستخدم عادة في إدخال النصوص الكبيرة مثل نص رسالة أو النصوص الكبيرة التي تكتب في عدد من السطور ليس في سطر واحد فقط ، كما في Textbox و لها الوسم الخاص بها في النماذج و هو <textarea>

- Name : أسم الأداة .
- Rows : و هي ارتفاع المربع بالصفوف ، أي صفوف الكتابة أو السطور
- Cols : و هي عرض المربع بالأعمدة ، أي أعمدة الكلمات



٥. الزر (button) :

و يعتبر من أهم أدوات النماذج ، فبواسطة الزر يتم تنفيذ الأوامر من قبل المستخدم ، فبعد كتابة النموذج أو ملئه ، يقوم بالضغط على الزر ليتم الإرسال ، و الزر له ثلاث أنواع :

- زر الإرسال (Submit) : و يكون مجهز لإرسال البيانات من النموذج غلى الصفحة الثانية مباشرة ، مثال السطر الأول في الصورة.
- زر المسح (Reset) : و يقوم بإعادة النموذج للوضع الافتراضي و مسح أي بيانات تم كتابتها، مثال السطر الثاني في الصورة.
- زر عادي (Normal Button) : و يقوم بتنفيذ أمر موجه أليه أو مكتوب تحته ، أي أمر، مثال السطر الثالث في الصورة.

```
<input type="button" value="Button" name="normal">
<input type="Reset" value="Reset" name="Reset">
<input type="submit" value="submit" name="submit">
```

- Name : أسم الأداة .
- Type : و هو نوع الـ Input أي تعرفه على انه زر من أي نوع من الثلاث أنواع.
- Value : و هي النص الذي يتم كتابته على الزر.



٦. مربع الاختيار الواحد (radio) :

و يستخدم للاختيار من بين مجموعة من الخيارات ، و لكن يتم اختيار خيار واحد فقط من مجموعة الخيارات المقدمة إلى العميل أو المستخدم في النموذج ، بشرط أن يكون بنفس الاسم ، بمعنى أنك تقدم إلى المستخدم مثلا نموذج يكون فيه مجموعة اختيارات للون المفضل من ثلاث ألوان ، سوف تضيف هذه الأداة ٣ مرات ، لها أسم واحد و لنفترض VColor و لكن تختلف القيمة فالأول تكون قيمته Red و الثاني تكون قيمته Green و الثالث تكون قيمته Black ، كما بالمثال التالي :

```
<input type="radio" value="red" name="vcolor" checked >أحمر<br>
<input type="radio" value="green" name="vcolor">أخضر<br>
<input type="radio" value="black" name="vcolor">أسمر</p>
```

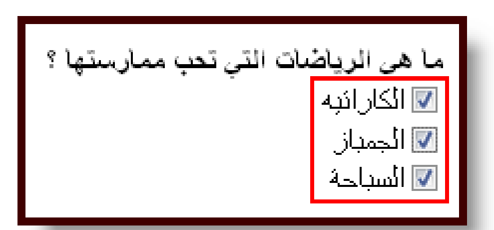


- Name : أسم مجموعة الاختيارات، لاحظ تسميه ال٣ اختيارات بنفس الاسم vcolor.
- Type : و هو نوع الـ radio أي تعرفه على انه أداة اختيار واحد من متعدد.
- Value : و هي عبارة عن القيمة الممثلة للاختيار.
- إضافات : مجموعة الاختيارات هنا تتكون من ثلاث عناصر ، إذا أردت أن تحدد عنصر مثلا ليكون محدداً افتراضيا ، فقم بوضع كلمه checked أي محدد قبل إغلاق الوسم أو الـ TAG.

٧. مربع الاختيار المتعدد (checkbox) :

و هذه الأداة من الأدوات الهامة هي الأخرى في النماذج و تستخدم عندما تحتاج أن تعطي للزائر أو المستخدم أكثر من خيار ليختارهم كلهم أو يختار أي مجموعه منهم أو أحدها ، فلا يشترط تحديد واحده فقط كما في الأداة السابقة، يكون لكل أداة الاسم الخاص بها ، و يتم تحديد قيمة للأداة و ترسل هذه القيمة عند إرسال النموذج و لكن في حالة تحديد هذه الأداة فقط ، تذكر في حالة تحديد الأداة فقط ، فالأداة التي يتم تحديدها ترسل قيمها، مثلا لديك نموذج تحب تسأل المستخدم فيه عن الرياضات التي يمارسها ، تقوم بإدراج ٣ أدوات كل واحده لها أسمها و قيمتها كما بالصورة:

```
<input type="checkbox" name="karate" value="on">الكاراتيه<br>
<input type="checkbox" name="gymnastic" value="on">الجمباز<br>
<input type="checkbox" name="swimming" value="on">السباحة</p>
```



- Name : أسم الأداة الواحدة.
- Type : و هو نوع الـ checkbox أي تعرفه على انه أداة اختيار متعدد.
- Value : و هي القيمة التي ترسل عند تحديد الأداة.
- إضافات : كما هو الأمر في radio إذا أردت تحديد أداة من هذه الأدوات من هذا النوع بشكل افتراضي فقم بكتابة الكلمة checked قبل إغلاق الوسم أو الـ Tag الخاص بالأداة التي تريد تحديدها.

```
<select size="1" name="country" >
<option value="EG" selected >مصر</option>
<option value="LY">ليبيا</option>
<option value="LB">لبنان</option>
<option value="JO">الأردن</option>
<option value="SY">سوريا</option>
<option value="IQ">العراق</option>
<option value="KW">الكويت</option>
<option value="SA">السعودية</option>
<option value="YE">اليمن</option>
<option value="OM">عمان</option>
<option value="PS">فلسطين</option>
<option value="AE">الإمارات</option>
<option value="BH">البحرين</option>
<option value="QA">قطر</option>
</select>
```

٨. القائمة المنسدلة (select) :

و هي أداة تستخدم للاختيار من متعدد ، إما اختيار واحد فقط أو عدة اختيارات ، تجدها كثيراً لتحديد البلد التي أنت منها في نماذج الشراء مثلا ، فيستحيل أن تستخدم أكثر من ١٠٠ radio في صفحة واحده ، من حيث الإمكان فممكن و لكن مظهر الصفحة لكن يكون

جيداً ، و لذلك كانت تلك الأداة هي الحل الأمثل لذلك، الأداة لها الوسم الخاص بها و ذلك أن تحتها أكثر من وسم كذلك، لاحظ الصورة.

حدد بلدك ؟
مصر
مصر
ليبيا
لبنان
الأردن
سوريا
العراق
الكويت
السعودية
اليمن
عمان
فلسطين
الإمارات
البحرين
قطر

- الوسم الرئيسي select :

و هو الوسم الذي يقوم بإنشاء الأداة و لكن فارغة من الاختيارات ، Name : هو أسم الأداة ، و Size هو ارتفاع الأداة بالصف ، أي أنك لو أدخلت ٢ لتم عرض صفين بشكل افتراضي في الأداة عند فتح الصفحة.

- الوسم الفرعي option :

أي اختيار ، لاحظ عدد الاختيارات هي مجموعة من الدول العشوائية العربية كل دوله في وسم خاص ، Value:هي قيمة الاختيار ، و يتم كتابة القيمة المعروضة للمستخدم في منتصف الوسم ، لتحديد عنصر بشكل افتراضي قم بكتابة selected قبل إنهاء بداية الوسم كما هو الحل مع التحديد "مصر" .

و بهذا أكون قد انتهيت من وصف الأجزاء المهمة العامة الخاصة بالأدوات المهمة بالنماذج و التي يكثر أستخدمها مع برمج PHP و مبرمج أي لغة تطويريه للويب، أرجو أن تكون هذه النقاط واضحة و إلا أرجو أن تراجعها حتى تكون مفهومة جيداً فما يلي معتمد على ما سبق إن شاء الله .

- **النماذج و PHP :**

لقد شرحت فيما سبق النموذج و مم يتكون و الأدوات التي تستطيع أن تستخدمها في النموذج لتخدم كل الاحتياجات الخاصة بالمستخدم الذي يستخدم هذا النموذج ، و التي تسهل عليه إدخال و البيانات، بعد وضع النموذج و وضع الأدوات المناسبة لك فإنك بذلك تكون قد انتهيت من نصف المطلوب تقريباً ، حيث أنك أعددت وسيلة الإرسال بقي أن تقوم بإعداد وسيلة استقبال هذه البيانات ، و في PHP من السهل جداً استقبال هذه البيانات ، حيث أن جميع البيانات التي تدخل في النموذج في أي نوع من أنواع الأدوات و بعد تحويلها لصفحة الـ PHP يتم تحويل تلك البيانات إلى عناصر في مصفوفة واحده ، حسب نوع الإرسال ، فإذا كانت طريقة نقل البيانات بالطريقة GET فإن البيانات الموجودة تتحول إلى المصفوفة $_GET و إذا كانت طريقة الإرسال POST فإنها تتحول إلى المصفوفة $_POST ، و هاتين المصفوفتين عبارة عن مصفوفتين ثابتين من ضمن مجموعة من المصفوفات الثابتة في PHP .

إذا يمكنك استخراج البيانات من صفحة النموذج غلى صفحة الـ PHP عن طريق هاتين المصفوفتين ، فجميع البيانات يتم تحولها بطريقة معينه ، حيث أن أسم الأداة يكون مفتاح العنصر و قيمة الأداة تكون هي قيمة العنصر في المصفوفة.

فمثلا لو لديك نموذج يحتوي على مربع نص أسمه name و كان النموذج الموجود فيه ينقل البيانات بالطريقة POST و قمت بإدخال أسمك مثلا في المربع و قمت بعمل إرسال (Submit) ، فإنه يمكنك الحصول على قيمة مربع النص من الصفحة السابقة في المصفوفة $_POST بهذا الشكل ($_POST["name"]) ، و لو أنك قمت بعرض قيمتها لعرض الاسم الخاص بك الذي كتبته في النموذج عند الإرسال.

مثال :

في هذا المثال سوف أقوم بعمل نموذج بسيط مثلا يطلب (الاسم الأول ، الاسم الأخير ، البلد ، النوع ، الهوايات ، ملاحظات) ، و هذا في صفحة أسمها form.html لاحظ الصور.

في الصورة الأولى و هي صفحة النموذج form.html تم إضافة الأدوات و في الصورة الثانية و هي صفحة الاستقبال send.php.

الصفحتين موجودتين في مجلد الأمثلة ، في الملفات المرفقة مع الكتاب ، أرجو أن تحاول تنفيذ المثال دون الرجوع إلى الكود.



الاسم : محيي الباز
البلد : مصر
النوع : ذكر
الهوايات : كاراتيه
الملاحظات : ملاحظات من نموذج الإرسال